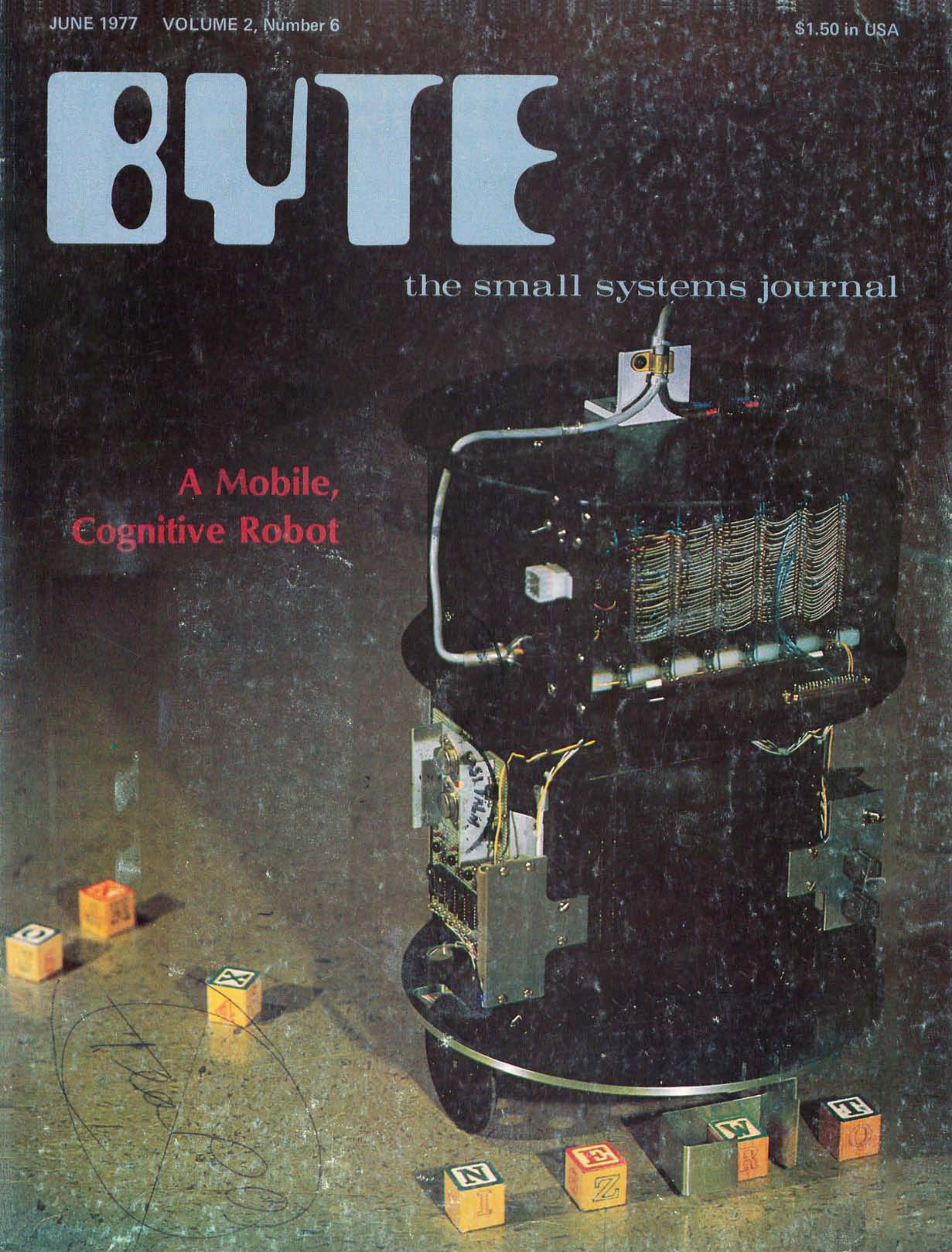
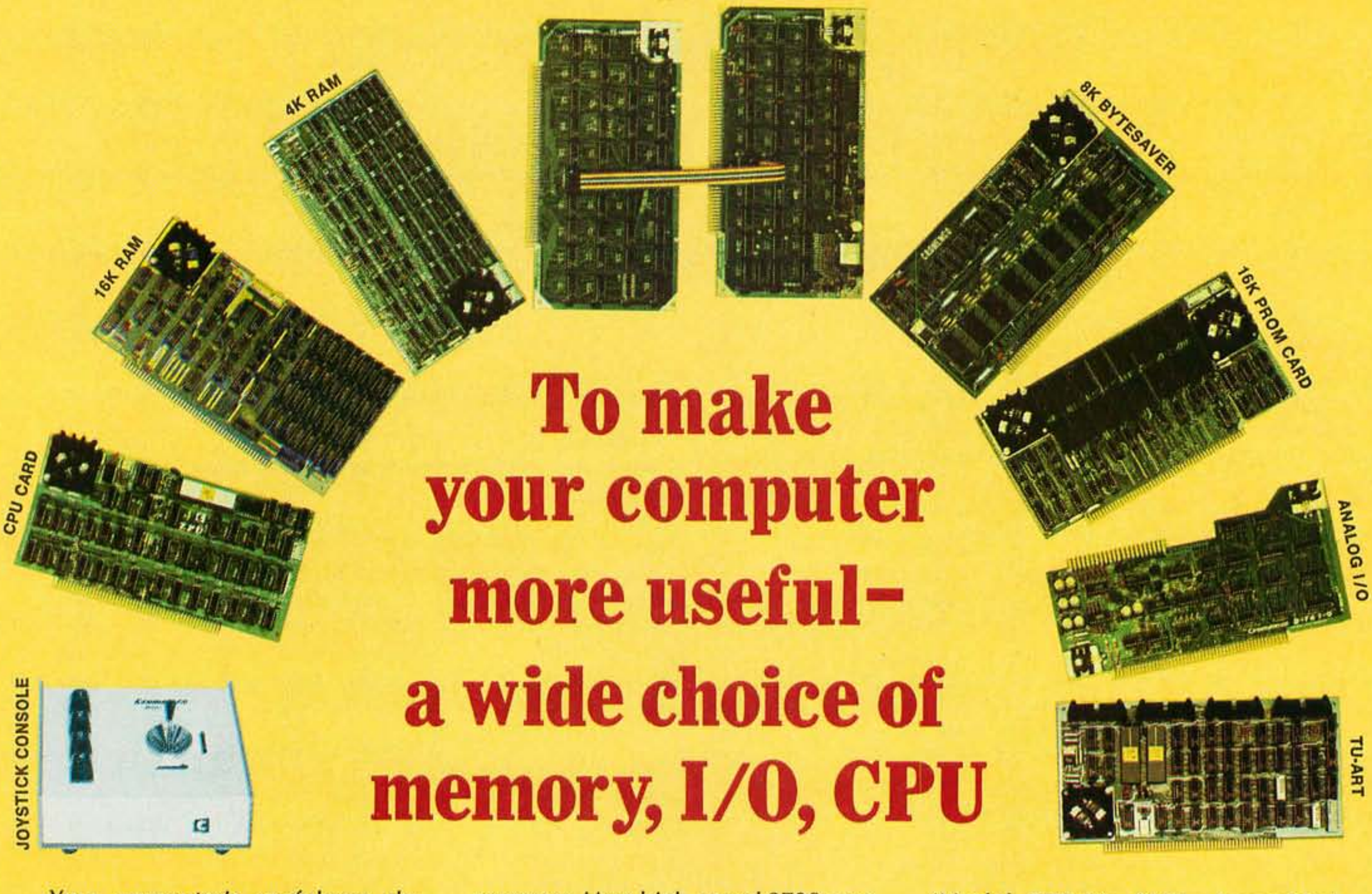
JUNE 1977 VOLUME 2, Number 6 $1.50 in USA

BYTE

the small systems journal

A Mobile, Cognitive Robot





To make your computer more useful– a wide choice of memory, I/O, CPU

Your computer's usefulness depends on the capability of its CPU, memories, and I/O interfaces, right?

So here's a broad line of truly useful computer products that lets you do interesting things with your Cromemco Z-1 and Z-2 computers. And with your S-100-compatible Altairs and IMSAIs, too.

CPU

• Z-80 MICROPROCESSOR CARD. The most advanced μP card available. Forms the heart of our Z-1 and Z-2 systems. Also a direct replacement for Altair/IMSAI CPUs. Has 4-MHz clock rate and the power of the Z-80 μP chip. Kit (Model ZPU-K): $295. Assembled (Model ZPU-W): $395.

MEMORIES

• 16K RAM. The fastest available. Also has bank-select feature. Kit (Model 16KZ-K): $495. Assembled (Model 16KZ-W): $795.

• 4K RAM. Bank-select allows expansion to 8 banks of 64K bytes each. Kit (Model 4KZ-K): $195. Assembled (Model 4KZ-W): $295.

• THE BYTESAVER — an 8K capacity PROM card *with integral programmer*. Uses high-speed 2708 erasable PROMs. A must for all computers. Will load 8K BASIC into RAM in less than a second. Kit (Model BSK-0): $145. Assembled (Model BSW-0): $245.

• 16K CAPACITY PROM CARD. Capacity for up to 16K of high-speed 2708 erasable PROM. Kit (Model 16KPR-K): $145. Assembled (Model 16KPR-W): $245.

I/O INTERFACES

• FAST 7-CHANNEL DIGITAL-ANALOG I/O. Extremely useful board with 7 A/D channels and 7 D/A channels. Also one 8-bit parallel I/O channel. Kit (Model D + 7A-K): $145. Assembled (Model D + 7A-W): $245.

• TV DAZZLER. Color graphics interface. Lets you use color TV as full-color graphics terminal. Kit (Model CGI-K): $215. Assembled (Model CGI-W): $350.

• DIGITAL INTERFACE (OUR NEW TU-ART). Interfaces with teletype, CRT terminals, line printers, etc. Has not one but *two serial I/O ports* and *two 8-bit parallel I/O ports* as well as 10 on-board interval timers. Kit (Model TRT-K): $195. Assembled (Model TRT-W): $295.

• JOYSTICK. A console that lets you input physical position data with above Model D + 7 A/D card. For games, process control, etc. Contains speaker for sound effects. Kit (Model JS-1-K): $65. Assembled (Model JS-1-W): $95.

PROFESSIONAL QUALITY

You get first-class quality with Cromemco.

Here are actual quotes from articles by independent experts: "The Cromemco boards are absolutely beautiful" . . . "The BYTESAVER is tremendous" . . . "Construction of Cromemco I/O and joystick are outstanding" . . . "Cromemco peripherals ran with no trouble whatsoever."

Everyone agrees. Cromemco is tops.

STORES/MAIL

So count on Cromemco. Look into these Cromemco products at your store. Or order by mail from the factory.

We wish you pleasure and success with your computer.

Cromemco
i n c o r p o r a t e d
Specialists in computers and peripherals
2432 CHARLESTON RD., MOUNTAIN VIEW, CA 94043 • (415) 964-7400

Circle 41 on inquiry card.

In the Queue

JUNE 1977 BYTE
Volume 2 Number 6

Foreground

Background

Nucleus

BYTE is published monthly by BYTE Publications Inc, 70 Main St, Peterborough NH 03458. Address all mail except subscriptions to above address: phone (603) 924-7217. Address all editorial correspondence to the editor at the above address. Unacceptable manuscripts will be returned if accompanied by sufficient first class postage. Not responsible for lost manuscripts or photos. Opinions expressed by the authors are not necessarily those of BYTE. Address all subscriptions, change of address, Form 3579, and fulfillment complaints to BYTE Subscriptions, PO Box 361, Arlington MA 02174; phone (617) 646-4329.
Second class postage paid at Peterborough NH 03458 and at additional mailing offices.
Subscriptions are $12 for one year, $22 for two years, and $32 for three years in the USA and its possessions. Add $5.50 per year for subscriptions to Canada and Mexico. For air delivery to western Europe, and for surface delivery worldwide, $25 for a one year subscription only. Worldwide air delivery available at additional rates. Please see subscription card.
Single copy price is $1.50 in the USA and its possessions, $2 in Canada and Mexico, and $3 elsewhere.
Foreign subscriptions and sales should be remitted in United States funds. Printed in United States of America.

page 46

In This BYTE

Some uses of a microprocessor involve the connection to the outside world through an analog interface. When fooling around with such projects from music generation to robotic control, however, it quickly becomes necessary to have a large number of inexpensive real world interfaces. To help point you in the right directions Douglas R Kraul supplied an article on **Designing Multichannel Analog Interfaces.**

In the past, readers have seen some interest expressed in the concepts of robotics, the use of small computers as the brains of mobile automated mechanisms. Robots have long been fancied in science fiction literature and cinema, but only rarely have people taken any practical steps towards a "real" robot as opposed to paper romanticisms or stage dummies. One of those rare cases is that provided by Ralph Hollis and his associate Dennis Toms, both of whom are physicists at the University of Colorado, Duane Physical Laboratory, Boulder CO. Ralph has been pursuing the design of practical robots as an avocation since 1957, and lately has progressed to the point of a working mobile computer system called Newt, whose picture provides the theme of this month's cover. Turn to Ralph's article, **Newt: A Mobile, Cognitive Robot** for essential background information on contemporary robot design philosophies.

Hard copy is a most useful output, but it tends to be somewhat expensive. Dan Fylstra shows one very attractive option in his article on **Interfacing the IBM Selectric Keyboard Printer.** Dan purchased a used print mechanism late in 1976, and since then has successfully interfaced the device to his KIM-1 system. Readers interested in using these printers (which are available in significant numbers on surplus markets) will find Dan's article an essential guide to the art.

How can hardware be used to accomplish the details of **Interfacing With an Analog World?** Turn to author Joseph Carr's second part of a two part series to find out some of the details of basic conversion circuits which use the outputs of sensors and preamplifiers discussed in last month's article.

Much of the software that is available on the market today is available on paper tape so as to be easily read into your microprocessor. The problem is that most common paper tape readers are so slow that it seems to take forever to read a large program into memory. In the article **Come Fly With KIM**, Rick Simpson introduces us to a solution to this speed problem: the Fly Reader, which he uses with MOS Technology's KIM-1.

Now that you've got the hardware built, how do you run it? Ken Welles answers this question in **Software for the Economy Floppy Disk.** His previous article (February 1977 BYTE, page 34) described how to construct an inexpensive floppy disk with minimal hardware. This month he provides a series of subroutines to run it, which could easily be expanded into a complete floppy disk operating system.

Last month in the first part of his article **Artificial Intelligence, An Evolutionary Idea,** Michael Wimble introduced us to the use of a simulated evolution technique by which it was possible for a program to alter itself and reshape its responses as a direct result of an outside stimulus. This month in **Part 2: Implementation,** Mr Wimble details how the computer experimenter can implement this type of program on any small computer system.

To many people the concept of assembly language is that of the fundamental language of the computer next to machine language. However, each particular assembly language command must be broken down into a series of simpler command sequences. These commands are known as microinstructions. In his article, **An Introduction to Microprogramming,** S M Quek describes how the concept of microinstructions is a great benefit to the user of a computer, allowing the easy change of basic instructions.

In previous issues Michael McNatt has shown us the availability of Baudot teleprinters and the ways in which they can be interfaced with your microprocessor. In his concluding article, **A Guide to Baudot Machines: Part 3, A Teleprinter Test Circuit,** he describes a test circuit that can be used for generating Baudot characters for alignment and adjustment purposes.

One Sol-20 equals three computers.

To do real work with any computer, big or small, it takes a complete system. That's one of the nice things about the Sol-20. It was built from the ground-up as the heart of three fixed price computer systems with all the peripheral gear and software included to get you up and on the air.

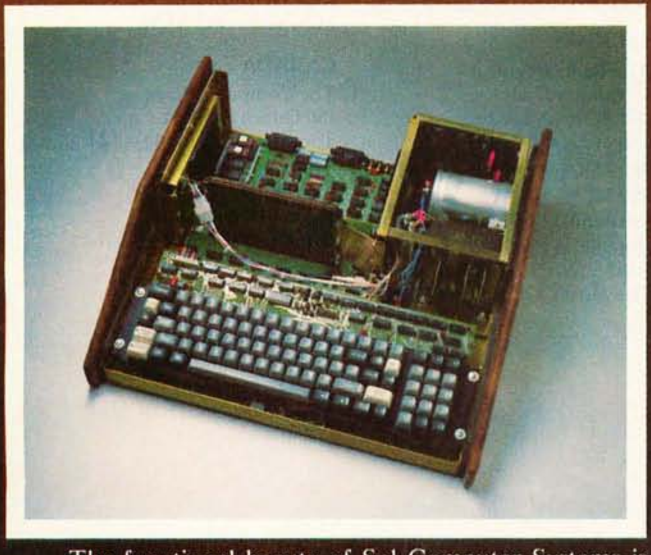
Sol System I costs just $1649 in kit form or $2129 fully burned in and tested. Here's what you get: a Sol-20 with the SOLOS personality module for stand alone computer power, an 8192 word memory, a 12" TV/video monitor, a cassette recorder with BASIC software tape and all necessary cables.

Sol System II has the same equipment plus a larger

capacity 16,384 word memory. It sells for $1883 in kit form; $2283 fully assembled.

For even more demanding tasks, Sol System III features Sol-20/SOLOS, a 32,768 word memory, the video monitor, Helios II Disk Memory System and DISK BASIC Diskette. Price, $4237 in kit form, $5037 fully assembled and tested.

And remember, though we call these small or personal computer systems, they have more power per dollar than anything ever offered. They provide performance comparable with mini-computer systems priced thousands of dollars more.



The functional beauty of Sol Computer Systems is more than skin deep. A look inside reveals a simple elegance of design and sturdy construction.

The Small Computer Catalog for the rest of the real computer system story.

Visit your local computer store for a copy of our fully illustrated 22 page catalog. Or you may write or call us if more convenient. Please address Processor Technology, Box B, 6200 Hollis Street, Emeryville, CA 94608. (415) 652-8080.

Processor Technology Corporation

The Software Dilemma:

Editorial

How is it possible to simultaneously make software widely available (and low priced), yet reward the producers of good software with adequate compensation for their efforts?

By Carl Helmers

This editorial consists of the text of a paper delivered at the First West Coast Computer Faire in April of this year.

Conventional wisdom has it that proprietary software must come at extremely high prices, commensurate with concentrated work on the part of a small number of dedicated and thoughtful programmers. After all, this wisdom has it, we'll only sell a few copies of package X anyway, so why not keep a tight lid on it and charge as much as possible?

This conventional wisdom has worked well in the past, when the typical computer system might cost upwards of $10,000 or $100,000. But when the typical computer system comes in at a price on the order of $1000, paying prices which are of this same order of magnitude for software packages is not a very likely move on the part of the individual purchaser with his or her personal budget.

In the personal computing field we are participating in a market phenomenon characterized by a change from the situation which supports the conventional software wisdom, to a new situation which has its own characteristics. More and more people are getting into the swing of things with computer use, and thus more and more people have needs which can and should be filled by specialized software products. Where computers are concerned, when we talk about a 100,000+ person active individual user market as we do today, we are for the first time talking about the potential for mass marketing of software in ways unheard of in the conventional wisdom of computing. Establishing a new "conventional wisdom" is clearly required; as a step toward that goal, this paper provides a survey of the prospects for mass marketing of software, and a solution of the software dilemma posed above.

Let's Draw Some Parallels: Woodworking

Like many individuals, I dabble a bit in the arts of crafting furniture. Suppose, for example, that I want to build a nice, neat contemporary rolltop desk for my study. As an individual with limited time available for such leisure crafts activity, I'd probably want to start with an existing design rather than working out all the details myself. In seeking the end product of a rolltop desk, I'd be in the same situation (as a wood craftsman) as the owner of a computer system desiring a compiler, assembler, application product or peripheral. I know in principle that rolltop desks exist and that in principle I could design then fabricate one, or use an existing one as a mental model with my own variations. But to save time and possible mistakes I might want to find some source of a "proven" design with detailed information on achieving the goal of a rolltop desk. Well, in the world of woodcrafting, as in the world of photography, the world of live steam model engines, or the world of backpacking, there are numerous sources of information including ready-made designs and techniques. I refer, of course, to books which are just published products with specific orientation or theme.

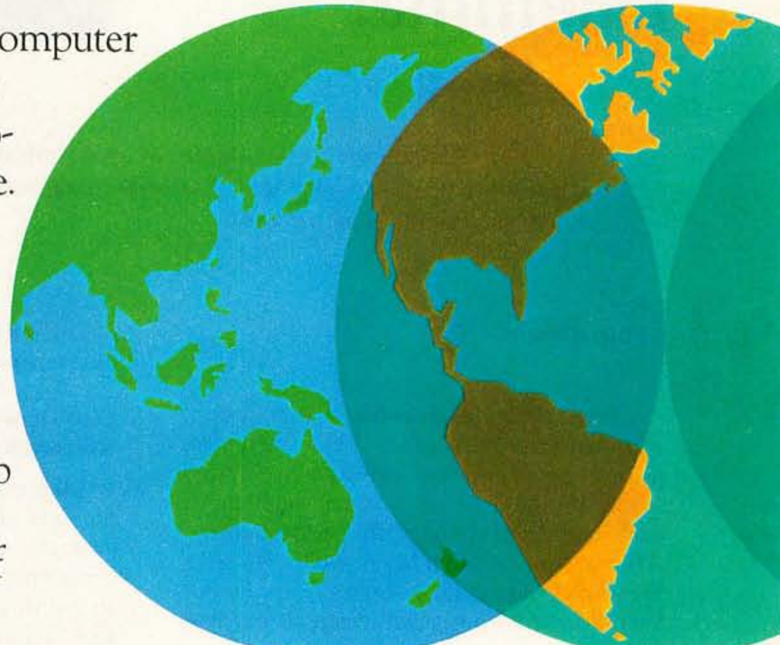
Similarly, when I have a computer system and I know that some neat language or software development tool exists, I also know that in principle I could write such a package myself using my own design or general design concepts taken from any

Continued on page 68

About those missing mailing wrappers and the May issue:

A strike at the printing plant was responsible for May BYTEs arriving late to subscribers and for May and June issues being mailed without the customary brown wrappers. The wrappers will be restored as soon as our printing situation is restored to normalcy. ▪

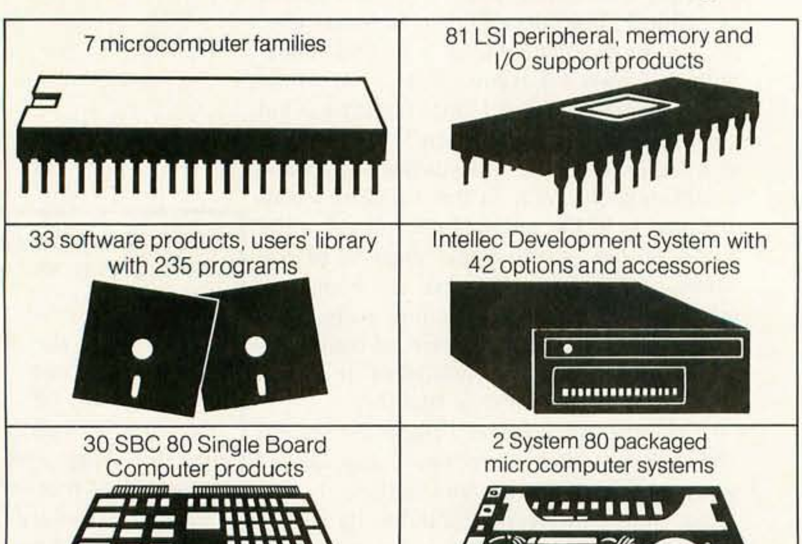
Intel delivers micro ahead of the fast



In 1971, Intel invented the microcomputer and quickly became the world's largest supplier of microcomputers and microcomputer support products. We still are.

Over the past six years we've invested millions of dollars to make the microcomputer even more useful and more economical. Today there are over 195 Intel® microcomputer hardware and software products available to help people like you keep ahead of costs, ahead of the competition and ahead of the fast changing world.

We're now offering seven microcomputer families. Including the newest high performance 8085 and the single chip 8748 with resident PROM. And 81 LSI peripheral, memory and I/O support circuits to help you cut design time, do more and get to market first. To reduce design time even further, choose one of our SBC80 Single Board Computers or System 80 packaged microcomputer systems.



But a wide selection of microcomputer components and systems is only half the story. We also provide programming support, including the PL/M high level microcomputer language to help you cut months off those big software development jobs. And Intellec® microcomputer development systems with ICE™ in-circuit emulation

Letters

AN APL LOVER'S STORY

Regarding the two letters by APL enthusiasts in your February 1977 issue, let me lend my voice to this group. While still in high school, I got my first taste of APL from a friend at IBM via a long-distance line to Detroit, and got access to Xerox Sigma 9 APL while a junior in college. Since then I've written a lot of APL code – including a 7 page pattern recognizer program (imagine what that would be in BASIC!) and about five pages of n-dimensional optimization routines, and now I am running IBM 5100 APL where I work. Needless to say, I think APL is the greatest thing since left-hand Turing machines.

I have just bought an ECD Micro-Mind computer (graphics) system with the explicit intent of buying 32 K of memory when the price comes down, and beg, borrow, buy, or if necessary, write an APL interpreter. (What about Tiny APL, analogous to Tiny BASICs being written now?) Since my computer will have graphics capability, I won't be interested in APL ROMs. (Might I suggest mnemonics, eg: $R for APL "rho," $QQ for APL "character quad.") I have used them "without hardly noticing." But in any case, put my vote in for APL, and I would be happy to hear from any APL enthusiasts.

Gregg Williams
3439 Southern, #7
Memphis TN 38111

SOME APL PERIPHERALS QUESTIONS

You can add my name to the list of those who would be interested in an APL character generator chip (Letters, February 1977). Like a lot of people who have used APL, I caught the bug, and have been disappointed that there is no software for the 8080 to support APL. Though I suspect it's only a matter of time.

A cost of $20 to $25 for a chip "feels" right to me. This would require about half the 500 buyers that Mr Montgomery postulates in his letter. I have a couple of warnings to add, though. First, a full upper and lower case keyboard is desirable, although it's certainly possible to use a spare control key to signal upper case and take care of the translation problem in software. Such keyboards don't seem to be as cheap and available as the surplus upper case ones. Second, I can attest to the fact that using little stickers on the keys to show you where the symbols are leads to a lot of frustration. Better to have keys imprinted with the letters and symbols. What would it cost to have sets of these made up for distribution with the character generators?

I'm convinced that there's a real market for a "small" (not tiny) APL interpreter. The word would spread fast to those who haven't had a chance to use it and are putting up with the inadequacies of BASIC (mainly the size of source programs) without much complaint.

James C Wilson
Ketron Inc
3250 Wing St #402
San Diego CA 92110

We know of one interpreter which is nearly complete for the TMS-9900, plus several 8080 versions. Watch future BYTEs for some fairly extensive APL information. Articles are now in preparation concerning APL interpreter design, use of APL, etc.

ON AUTOMATED BAROMETERS AND OBTAINING MERCURY

I was very interested in Mr Firth's article on weather predictions (December 1976 BYTE, page 62), for there have been very few articles in BYTE on getting a computer to do things other than to play Star Trek. I was especially interested in his idea about getting barometric readings into a computer. Being a chemistry student I ran across an article in the *Journal of Chemical Education* (October 1976). Although the barometer in the *Journal* is a little different, it's basically the same as Mr Firth's.

Also, in Mr Firth's article he mentions that you need a quarter pound of mercury, and many lab supply houses will not sell you mercury for any reason without a company's purchase order. Another way of obtaining the mercury, although it may take some time, is to remove the mercury switches from old washing machines. The mercury is not very pure, but it can be cleaned up somewhat by passing it through a pin hole in filter paper. The mercury obtained by this method is good enough to be used in the barometer. This may seem a lot of work to go through, but it sure beats paying $13 for a quarter pound of mercury that you will probably use a quarter of.

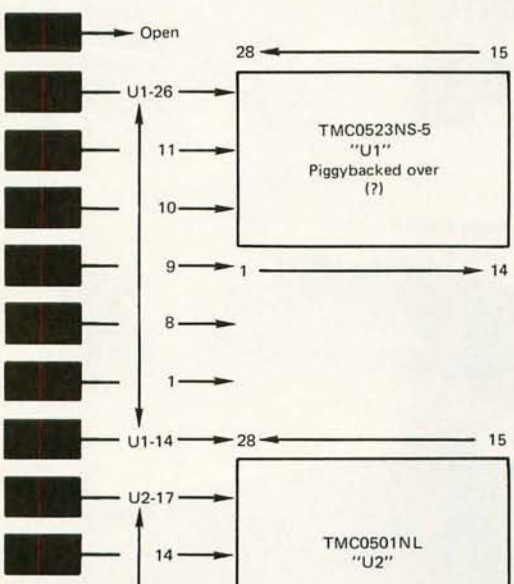
D Pasken
23 Farview Cir
Camillus NY 13031

Mr Pasken enclosed a Xerox shot of the article he mentions, which can be found on page 670 of the October 1976 *issue of the* Journal of Chemical Education. *The design is by John T Viola and William E McDermott of the US Air Force Academy. The design, in detail, is a recording manometer. Reference is made to a paper, circa 1953, by H T Svec and D S Gibbs (in* Rev Sci Instr, **24**, *202, 1953). The paper also gives a reference to a source of wire for the measurement: 28 gauge bare nickel chrome wire cable cord manufactured by Consolidated Companies, Chicago IL.*

SOME SR-51 CALCULATOR INTERFACE INFO

I read with interest Ralph Getsla's (Letters of January 1977 BYTE) request for information on interfacing his SR-52. My interests are similar, only my outlook evolves around the use of the SR-51 A terminal strip that can be seen upon removal of the battery pack of the 51. My plan here is to interface the 51 to the modified TV typewriter terminal that I am presently in the completion stages of building, I was able, so far, to track these lines back to their source by the use of a hand held flashlight after disassembly of the 51 (see interface diagram). If I can get any information on the two chips in question I believe I would be in business.

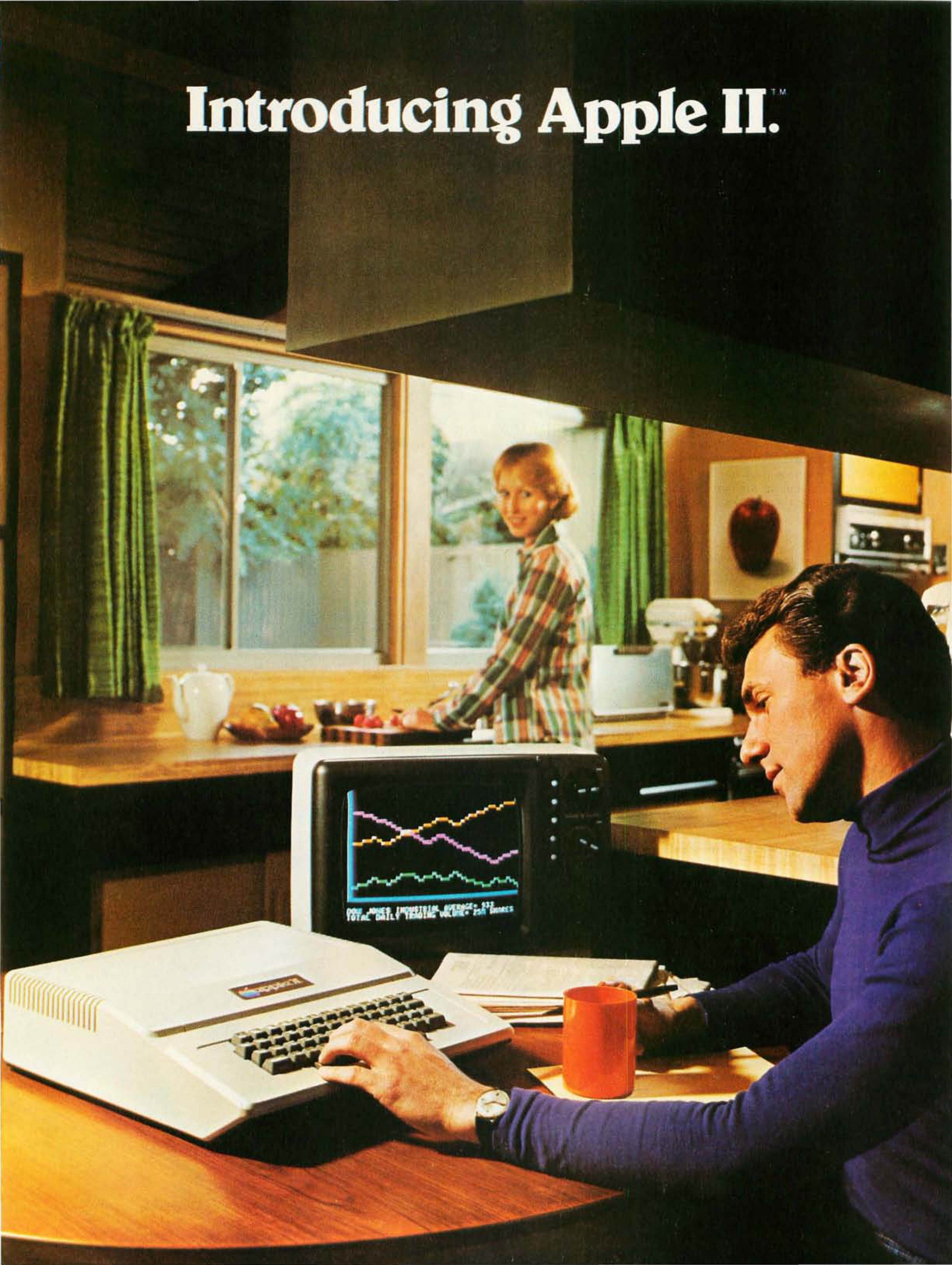
William D Lewis
469 Heatherbray Ct
San Jose CA 94136



PS: Could this interface be the same as the SR-52? I will be writing Texas Instruments for any information they will be able to give me on these devices. If not, I will operate via the probe.

Who knows?

Continued on page 122


Introducing Apple II.™

You've just run out of excuses for not owning a personal computer.

Clear the kitchen table. Bring in the color TV. Plug in your new Apple II,* and connect any standard cassette recorder/player. Now you're ready for an evening of discovery in the new world of personal computers.

Only Apple II makes it that easy. It's a

complete, ready to use computer, not a kit. At $1298, it includes video graphics in 15 colors. It includes 8K bytes ROM and 4K bytes RAM—easily expandable to 48K bytes using 16K RAMs (see box). But you don't even need to know a RAM from a ROM to use and enjoy Apple II. For example, it's the first personal computer with a fast version of BASIC permanently stored in ROM. That means you can begin writing your own programs the first evening, even if you've had no previous computer experience.

The familiar typewriter-style keyboard makes it easy to enter your instructions. And your programs can be stored on—and retrieved from—audio cassettes, using the built-in cassette interface, so you can swap with other Apple II users.

You can create dazzling color displays using the unique color graphics commands in Apple BASIC. Write simple programs to display beautiful kaleidoscopic designs. Or invent your own games. Games like PONG—using the game paddles supplied. You can even add the dimension of sound through Apple II's built-in speaker.

But Apple II is more than an advanced, infinitely flexible game machine. Use it to teach your children arithmetic, or spelling for instance. Apple II makes learning fun. Apple II can also manage household finances, chart the stock market or index recipes, record collections, even control your home environment.

Right now, we're finalizing a peripheral board that will slide into one of the eight available motherboard slots and enable you to compose music electronically. And there will be other peripherals announced soon to allow your Apple II to talk with another Apple II, or to interface to a printer or teletype.

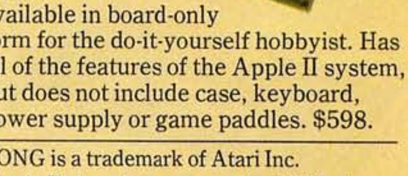
Apple II is designed to grow with you as your skill and experience with computers grows. It is the state of the art in personal computing today, and compatible upgrades and peripherals will keep Apple II in the forefront for years to come.

Write us today for our detailed brochure and order form. Or call us for the name and address of the Apple II dealer nearest you. (408) 996-1010. Apple Computer Inc., 20863 Stevens Creek Boulevard, Bldg. B3-C, Cupertino, California 95014.

Apple II™ is a completely self-contained computer system with BASIC in ROM, color graphics, ASCII keyboard, lightweight, efficient switching power supply and molded case. It is supplied with BASIC in ROM, up to 48K bytes of RAM, and with cassette tape, video and game I/O interfaces built-in. Also included are two game paddles and a demonstration cassette.

SPECIFICATIONS

- **Microprocessor:** 6502 (1 MHz).
- **Video Display:** Memory mapped, 5 modes—all Software-selectable:
  - Text—40 characters/line, 24 lines upper case.
  - Color graphics—40h x 48v, 15 colors
  - High-resolution graphics—280h x 192v; black, white, violet, green (12K RAM minimum required)
  - Both graphics modes can be selected to include 4 lines of text at the bottom of the display area.
  - Completely transparent memory access. All color generation done digitally.
- **Memory:** up to 48K bytes on-board RAM (4K supplied)
  - Uses either 4K or new 16K dynamic memory chips
  - Up to 12K ROM (8K supplied)
- **Software**
  - Fast extended BASIC in ROM with color graphics commands
  - Extensive monitor in ROM
- **I/O**
  - 1500 bps cassette interface
  - 8-slot motherboard
  - Apple game I/O connector
  - ASCII keyboard port
  - Speaker
  - Composite video output



Apple II is also available in board-only form for the do-it-yourself hobbyist. Has all of the features of the Apple II system, but does not include case, keyboard, power supply or game paddles. $598.

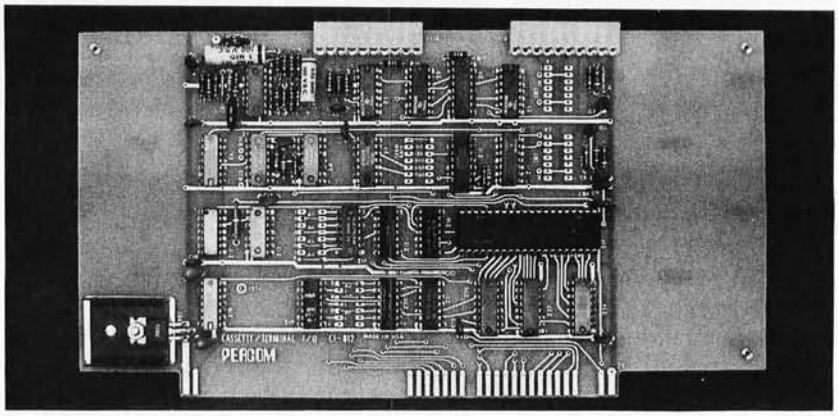
PONG is a trademark of Atari Inc.

*Apple II plugs into any standard TV using an inexpensive modulator (not supplied).

apple computer inc.™

Circle 272 on inquiry card.

What's New?

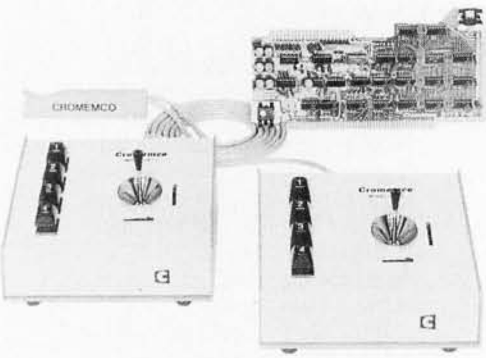


A New Audio Tape Cassette Driver for the Altair Bus

PerCom Data Company Inc, 4021 Windsor, Garland TX 75042, has just introduced a new version of Harold Mauch's Kansas City standard phase encoding audio signal interface board, a version which plugs directly into an Altair bus slot. Harold's design allows phase encoding with redundancy at 300 bps (Kansas City standard), 600 bps, 1200 bps and 2400 bps. In addition to the tape interface function, the CI-812 product also includes a companion RS-232 terminal interface with data rates from 300 to 9600 bps. The kit price of this board is $89.95, and an assembled version is $119.95.■

Circle 602 on inquiry card.

Games are More Fun with Action Inputs



This little truism can be confirmed by anyone who has implemented and played a space war with joystick control, or used joysticks for direct control of moving systems such as robots or other mechanical marvels. Cromemco, 2432 Charleston Rd, Mountain View CA 94043, has sent along this photo of the new model JS-1 joystick console adapters for their Altair compatible D+7A IO board. What you get is two independent game control boxes with two axis joysticks, four game function buttons, and loudspeaker outputs for aural effects (such as photon torpedo or phaser sounds). The kit form of the box (in foreground) is available for $65.■

Circle 601 on inquiry card.

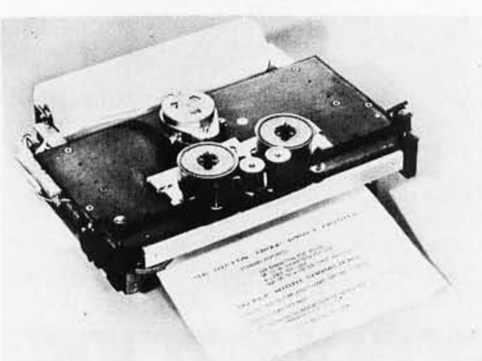
Inexpensive Wire Wrap Tools

OK Machine and Tool Corporation, 3455 Conner St, Bronx NY 10475, has come out with a unique product line of wire wrapping tools and accessories for the amateur electronics person. These products include manual and battery powered wire wrapping tools, precut and stripped wire, wire rolls, dual in line package sockets, and wire wrapping "kits." Of particular interest to people on a tight budget is a new low in prices for wire wrapping tools which are powered. The OK BW-630 battery powered wire wrap tool uses standard C size batteries and comes equipped with a bit and sleeve for wrapping AWG 30 wire for only $34.95 (less batteries). This is not a kluge, but a genuine wire wrap gun with positive indexing mechanism to return the bit to a well defined position after each wrap, and the usual "antibackforce" spring loading of the bit to prevent overwrapping. Both of these features are standard items on the industrial wrapping guns which have been used for years.■

Circle 604 on inquiry card.

Hard Copy That's Hard to Beat for Speed

The Digital Group, POB 6528, Denver CO 80206, has announced what is probably the fastest and widest line width matrix impact printer mechanism and electronics package yet to be marketed to personal computing users. In kit form (kit refers to the electronics, not the mechanics), prices for this printer start at $495. What you get is a fast 120 characters per second 5x7 dot matrix printer which gives 96 characters per line at 12 characters per inch pitch, and line spacing of six lines per inch.



Since it is a true impact printer mechanism intended for computer systems use, it will take up to four part forms and should prove most useful to business people for that reason. For the amateur computer person with software development in mind, the 120 character per second rate means listings of assemblies and compilations which take one twelfth the time of a 10 character per second Teletype, but at a price for the mechanism and its interface which is lower than the new cost of a Teletype! Other features of the OEM mechanism selected by the Digital Group include built-in ribbon reinkers for a total ribbon life of 10,000,000 characters, use of 8.5 inch (22 cm) wide standard roll, fanfold or sheet paper, an 8 bit parallel interface ready to plug into your computer's output port after you wire up the cable, and the option of double width characters. For those with idle curiosity, the inking life of 10,000,000 characters before replacement of ribbon corresponds to over 23 hours of flat-out printing, or 250,000 lines with 40 nonblank characters per line.■

Circle 603 on inquiry card.

Order your Apple II now.

Use this order form to get your Apple II fast. As a special offer for those who order now, we will include free a custom vinyl carrying case (a $50 value). And we will also pay shipping charges to anywhere in the continental United States.

Apple II Price List.

RAM Complement	Apple II System	Calif. Residents Add	Apple II Board-only	Calif. Residents Add
4K	$1,298.00	$ 84.37	$ 598.00	$ 38.87
8K	1,398.00	90.87	698.00	45.37
12K	1,498.00	97.37	798.00	51.87
16K	1,698.00	110.37	978.00	63.57
20K	1,778.00	115.57	1,078.00	70.07
24K	1,878.00	122.07	1,178.00	76.57
32K	2,158.00	140.27	1,458.00	94.77
36K	2,258.00	146.77	1,558.00	101.27
48K	2,638.00	171.47	1,938.00	125.97

Memory is offered at a 20% savings when ordered with the system–or board–as reflected in the prices above.

Additional RAM can be easily added-in at a later date as your needs develop.
One set 4K chips (4K bytes) $125
One set 16K chips (16K bytes) $600

Prices and specifications subject to change without notice.

apple computer inc.™
20863 Stevens Creek Blvd., B3-C
Cupertino, California 95014
(408) 996-1010

Order Form

☐ Please send me an Apple II System
☐ Board Only
with ____ K bytes of RAM (4K minimum) at $__________
California Residents add 6.5% tax __________
Total $__________
Name __________
Address __________
City __________
State __________ Zip __________
Phone __________
Shipping Address (if different)
____
____

☐ Cashier's check or money order enclosed.
(Please allow 2 additional weeks for personal checks.)
Please charge to my
☐ BankAmericard
☐ VISA
☐ Master Charge
Card Number __________
Expiration Date __________
Signature __________

Mail to: **Apple Computer Inc., 20863 Stevens Creek Blvd., B3-C, Cupertino, California 95014**

Designing Multichannel

Douglas R Kraul
4373 Ashwoody Trl
Atlanta GA 30319

Analog interfaces to and from the personal computer system can present a difficult dilemma to the small systems user: The analog interface usually is a very expensive proposition, especially if more than one input and one output are needed. Schemes like that suggested by Roger Frank *[page 70 of the May 1976 issue of BYTE]*, can greatly reduce hardware complexity, and thus cost, since much of the interface burden is left to the software of the computing system. Direct extension of this principle to the case of multiple input voltages and multiple output voltages can, however, result in a hardware cost that at the least rises linearly with the number of needed outputs and inputs. One alternative scheme requires an additional bit of input to the computer and one additional voltage comparator for each additional analog to digital input up to a total of 8. On the output side each additional voltage output leads to an additional 8 bit output port and an additional 8 bit digital to analog converter. This results in a situation where a many input, many output analog interface requires an inordinate amount of hardware, which means money to the user. (We should not kid ourselves by saying that large numbers of analog channels are rarely needed. Many worthy applications, like control of analog music synthesis, automated test facilities or control of robots would easily push the number of channels needed beyond the point of no return for the previously suggested schemes of interface.) Thus one must turn to a modified philosophy of interface design in order to meet the necessary goal of a less expensive analog interface.

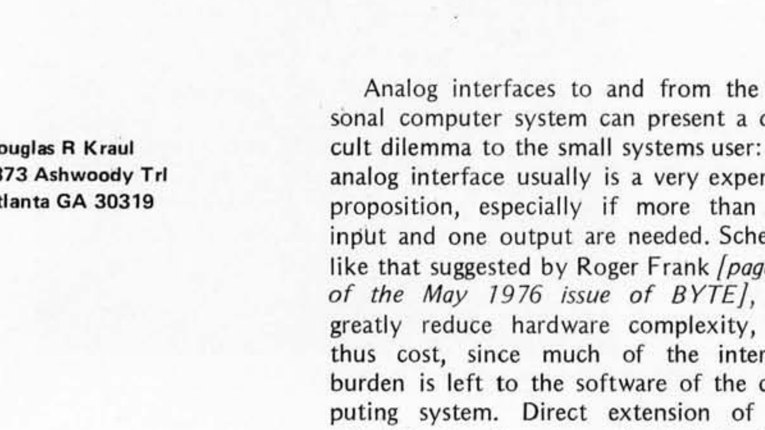


Figure 1: Block diagram symbolizing the hardware that is needed for a multiplexed analog to digital converter for eight inputs. The three outputs from IO port B select the analog channel. The output from the multiplexer is fed to the negative side of a voltage comparator. An analog output from a digital to analog converter is fed to the positive side of the comparator. When the analog value from the converter is greater than the value being tested, the voltage comparator will output a 1. Using successive approximations the input voltage can be determined.

Fortunately digital techniques provide us with a method of solution to the problem: time multiplexing. Time multiplexing is simply the process by which one device can be made to function as many logical devices. To the user these virtual devices appear as if they were full fledged dedicated devices. Thus our objective is to find some technique by which one analog to digital converter and one digital to analog converter can be made to function in many seemingly simultaneous conversions.

Basics of Time Multiplexed Interfaces

The basic principles are illustrated first for the analog to digital case. Figure 1 illustrates the hardware that allows multiplexed analog to digital conversions. An output port, A, from the computer is used to provide the necessary eight bits to drive the digital to analog converter (DAC). The output of the digital to analog converter is connected to the minus input of the voltage

Analog Interfaces

comparator. The output from the comparator provides an input to the computer by way of the most significant bit of an input port to the computer. This structure thus far is identical to the scheme proposed by Roger Frank. The difference is that the positive input to the comparator is no longer connected directly to the voltage to be converted. Rather the comparator is connected to the voltage to be converted by way of an analog multiplexer.

The analog multiplexer here is performing the needed function that allows one analog to digital converter to deal with many channels. A typical application might use an 8 to 1 multiplexer. Thus any one of the eight voltage inputs might become the voltage to be converted if the multiplexer selects it.

The selection is accomplished by a binary code applied to the select input of the multiplexer. In an 8 to 1 device the binary code 011 would pick the input labeled 3. The code which selects the input was set by the computer through an output port. For our 8 to 1 example three bits would be needed, possibly originating from the lower three bits of an 8 bit output port.

This change results in almost no change in the software that would service the analog to digital conversions. In fact, the only necessary modification is to preload the channel selection word, which chooses the voltage to be converted in the proper output port. Then the analog to digital conversion routine can be called.

Time Multiplexed Digital to Analog Conversion

Multiplexing of the analog to digital conversion really only solves half of the analog interface problem. The problem of economically generating multiple analog outputs from the converter still remains. We can apply the versatile analog multiplexer to solve this problem as well. (However, there are complications that can mask the simplicity of the method.)

The basic hardware of the multiplexed digital to analog converter output is shown in block form in figure 2. Once again the source of the digital to analog converter's word is an 8 bit output port from the computer. The output is now connected to multiplexed sample and hold circuits. Much like the multiplexer used in the analog to digital conversion system the multiplexed sample and holds connect the output of the digital to analog converter to the desired analog output which is chosen by the select inputs to the multiplexer. The difference between the plain analog multiplexer and

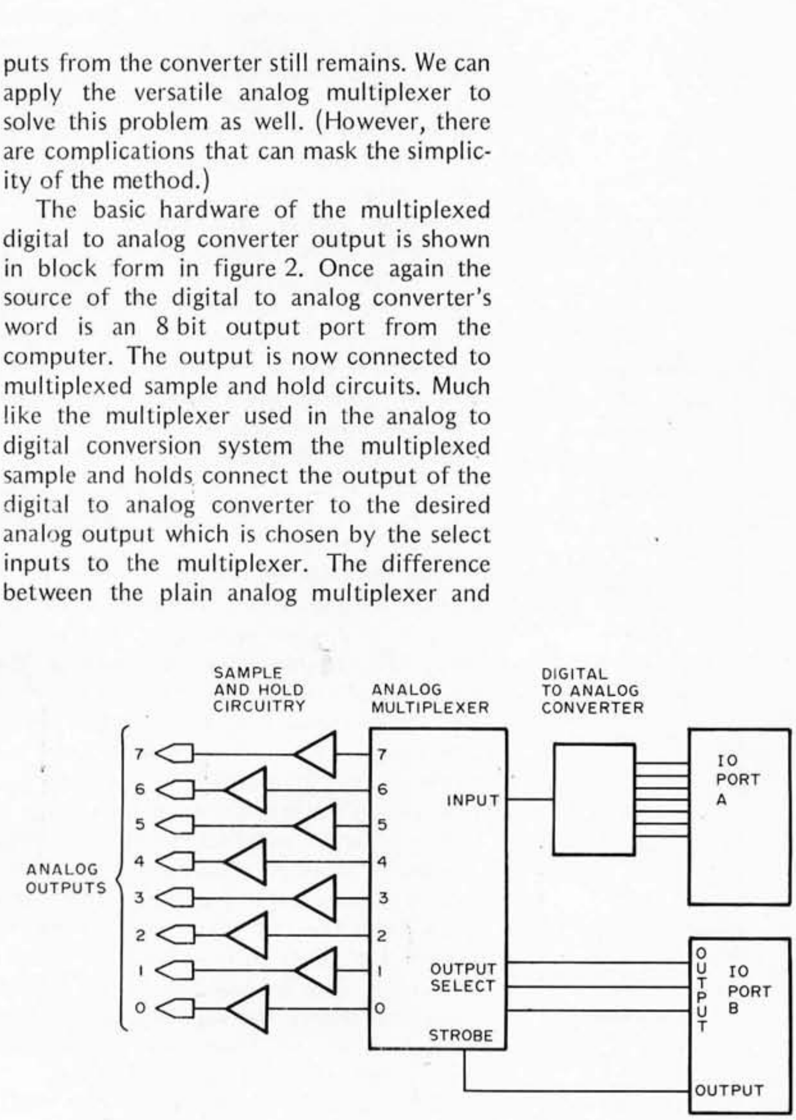


Figure 2: Block diagram of the basic hardware needed for a multiplexed digital to analog converter for eight channels of output. Three bits of output from IO port B select the channel that is to be used. When a strobe is enabled the chosen channel is activated. Each analog channel has a sample and hold circuit which must be updated periodically due to the leakage of the capacitor.

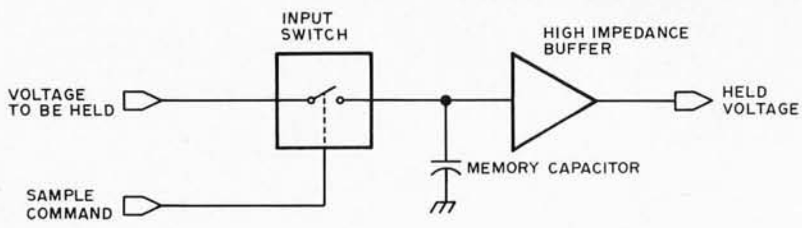


Figure 3: Functional schematic of a typical sample and hold circuit. The input switch can be mechanical or electronic. In the case of a multiplexed interface it is an analog switch.

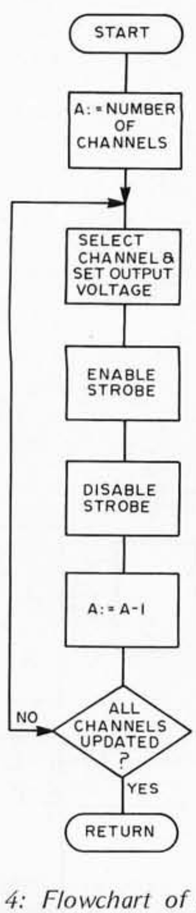


Figure 4: Flowchart of a typical IO driver for a digital to analog system. This program is called periodically to update the value that is being held by the sample and hold circuitry.

the multiplexed sample and hold is memory. The multiplexed sample and hold has the ability to remember the voltage that was applied to the desired output (for a while).

Before proceeding, some background information on sample and hold circuits in general will prove instructive. A functional schematic of a sample and hold circuit is shown in figure 3. The major components of the sample and hold circuit are an electronic input switch, a memory capacitor and a high impedance buffer. The sample and hold circuit works as follows: The voltage to be remembered is applied to the input. The switch is closed, allowing the voltage to be applied to the capacitor. The switch is then opened and the input voltage can now be changed because the output of the sample and hold now reflects the formally applied voltage. This discussion assumes ideal components. There are a number of error sources. The majority revolve around the memory capacitor.

The remembered voltage is stored as an electric charge in the memory capacitor. Because of this any variation of charge with time, ie: current, causes an error in the remembered voltage. This explains the need for a high input impedance in the buffer amplifier so that it doesn't drain away too much charge. A measure of a sample and hold circuit's ability to retain the voltage to within a certain percentage is its hold time. Another problem associated with the memory capacitor is acquisition time. This arises from the fact that charge cannot be delivered instantly to the capacitor. A finite amount of time is needed to deliver enough charge to change the capacitor to the new voltage. Thus we have two design parameters: hold time and acquisition time.

The multiplexed design is not much different from the principles outlined above. The sample switch is merely replaced by our friend, the analog multiplexer. The output to be changed is selected, the strobe then enabled (closing the switch) and after the acquisition time, disabled (opening the switch). The select word and the strobe will possibly originate from the lower four bits, for an 8 output system, of an output port from the computer.

This type of interface does represent a burden on the computer. The reason for this burden is the very finite hold time of the capacitor. No sample and hold circuit can retain its value forever. The time can be increased by using a larger capacitor, but a limit is reached because a larger capacitor leads to longer acquisition times. Thus, the sample and hold device must be updated periodically if the outputs are to remain accurate. This situation is not unlike that of dynamic memories which are effectively two state sample and hold circuits.

Use of the Multiplexed Digital to Analog System

Obviously this type of interface will require much more computer intervention. The software, though, is not difficult. A possible IO driver is flowcharted in figure 4. For an 8 output system an 8 entry data file is needed to hold the current output values. Periodically (perhaps cued by the interrupt system) with a period less than the hold time a routine is executed to update the eight outputs. The main loop of this routine consists of the following: The value of the output presently being updated is sent to the digital to analog converter interface. The output is then selected and strobed. This action then repeats until all eight outputs have been updated.

What Time Multiplexing Buys You

For the small systems user minimizing hardware is essential. The potential saving of a time multiplexed analog interface is high. The reason for this in general is the reduced hardware. Another not so apparent reason is the ease of expansion.

The most obvious hardware savings occurs in the digital to analog converter. This is because eight channels can be had for the cost of one converter, one 8 to 1 analog multiplexer, eight memory capacitors, eight output buffers, and two output ports. This is contrasted to the eight converters, and the eight output ports from the computer needed by the conventional brute force approach. On the input side the gain is not as obvious. Here we have replaced eight comparators with one comparator and a multiplexer. Both cases require an output port, an input port and a digital to analog converter. A check of prices reveals, though, that eight comparators cost more than one multiplexer.

The clincher is when one considers updating the system to more than eight channels. The output channel needs only to increase the number of sample and hold devices which is cheaper by factors of eight. The brute force scheme required one more converter and one more output port per additional channel. Table 1 compares the number of components needed for a conventional and a multiplexed system based on 16 channels.

The multiplexed analog to digital conversion process has an even more spectacular success. In the dedicated hardware approach each channel requires one comparator and one bit of input to the computer. The increment for the multiplexed approach is only one multiplexer per eight additions, and one bit of output. Table 2 compares the needed components, based on 16 channels. Thus overall we see that the multiplexed approach offers a multitude of hardware savings.

What Time Multiplexing Costs You

This design technique is typical of many that trade hardware for software. Obviously since we have taken so much out of the hardware, the software and computer efficiency will degrade. It is perhaps a truism that if the interface is designed intelligently these problems can be minimized.

The analog to digital interface suffers the least. The main problem here is the amount of time spent doing the conversion routine. If fast changing inputs or a multitude of moderate inputs are to be converted then the computer is severely loaded. However, many applications only require moderate conversion rates. Foremost of these are interfaces to human operators. Maximum conversion rate needed here is around 100 Hz. Typically, this is around 0.001 Hz to 0.1 Hz. Examples of this are the proportional controls in games and operator set parameters. Control signals in electronic music also fall in these categories. Thus this type of interface can work well in many typical applications if the rates and the numbers are not excessive.

Component	Dedicated	Multiplexed
Digital to analog converter	16	1
IO ports	16+16 bits	2
8 to 1 multiplexer	0	2

Table 1: Comparison of the amount of hardware that is needed for the direct method of digital to analog converter versus the multiplexed method of interfacing. The table is constructed for an interface consisting of 16 channels.

Component	Dedicated	Multiplexed
Digital to analog converter	1	1
IO ports	3	2
Voltage comparators	16	1
8 to 1 multiplexers	0	2

Table 2: Comparison of the amount of hardware needed for direct methods of analog to digital interfacing versus the multiplexed method of interfacing. The table is constructed for an interface consisting of 16 channels.

The digital to analog interface suffers from a similar situation. Here the main problems are extremely slow changing outputs or large numbers of fast outputs. Here again an analysis of likely applications reveals that a great number of output signals reside in the frequency spectrum between 1 Hz and 100 Hz. Sample and hold circuitry can be economically designed with hold times in the excess of one second. Also the whole refresh process can be made transparent to the main program if it is done under interrupt control by causing the update routine to be executed at the rate of the fastest output in response to the request of a programmable timer. As with the analog to

GLOSSARY

Acquisition Time: The time required for a sample and hold circuit to change from its previous value to its new value within a prescribed tolerance.

Analog Multiplexer: A solid state device that allows a multitude of connections to be accessed by a common line. The action is like an N position switch.

Comparator: An analog device whose output is logical 1 if the plus input is greater than the minus input and logical 0 if the situation is reversed.

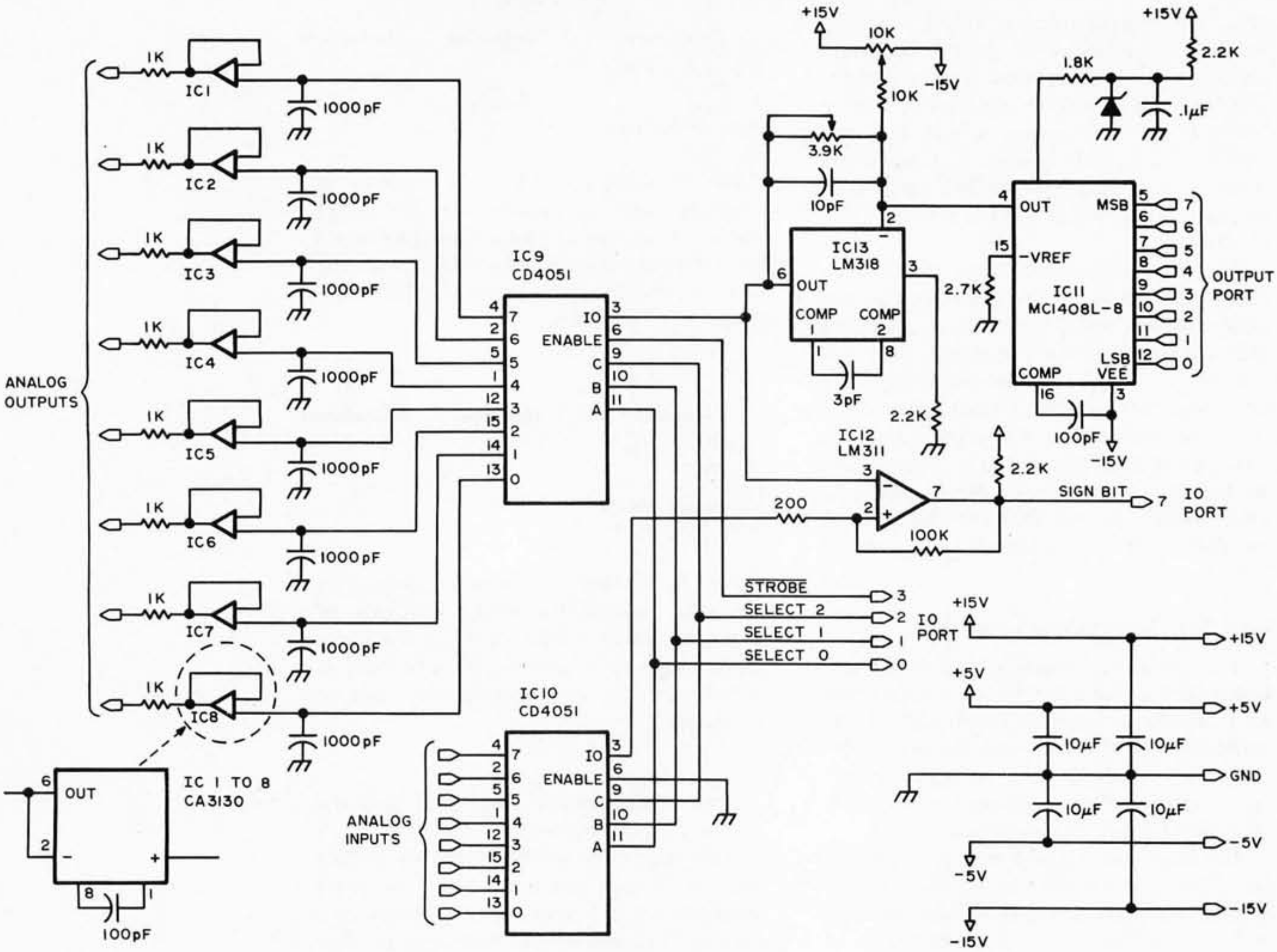
Digital to Analog Converter (DAC): A device whose output analog signal (current typically) is proportional to a digital word at its input.

High Input Impedance Buffer: A device whose input draws little current from any other devices connected to it. It is important in a sample and hold circuit since currents cause the voltage held in the memory capacitor to discharge.

Hold Time: Amount of time that passes before the output from a sample and hold circuit changes from the originally held value by a prescribed tolerance.

Sample and Hold: Analog memory device which stores a voltage as electrical charge in a capacitor.

Time Multiplexing: Process of combining several measurements for transmission over one signal path. In our case, this signal path is the IO port structure of a processor.



Number	Type	+5 V	GND	−15 V	+15 V	−5 V
1 to 8	CA3130	7				4
9	CD4051	16.	8			7
10	CD4051	16	8			7
11	MC1408L-8	13	2			
12	LM311		1	4	8	
13	LM318			4	8	

Figure 5: Schematic of the multiplexed analog to digital and digital to analog interface. The parts were mostly chosen for speed and cost considerations. The integrated circuits are mostly CMOS. All resistances are measured in ohms and all resistors are 1/4 W. Be sure to bypass each power pin with a 0.01 µF capacitor to help eliminate any stray spikes originating from power surges.

digital cases this type of interface works well with a moderate number of medium speed outputs or a multitude of low speed outputs. Note that outputs which can be changed at a rate up to that of the acquisition time (if only one channel is used) can be had by using fewer channels in the output program loop. Outputs that meet these requirements are some automatic testing signals, mechanical devices and control signals for electronic music.

To summarize, this type of interface burdens the computer system. Using it wisely for moderate types of signals does lead to a workable system. Conveniently these types of signals are quite prevalent in interfaces to humans, and many types of equipment.

A Complete Design Example

To illustrate these principles in more concrete terms a complete interface is described in figure 5. The 8 channel interface represents hardware well within the realm of most small systems users. It is optimized for signals from 0.1 Hz to around 100 Hz, though lower and higher rates are possible at the expense of efficiency. This type of interface is useful in interactive games, testing of equipment and electronic music. Parts used are neither exotic nor expensive.

Figure 5 contains the complete circuit. The following comments on component selection are in order. The digital to analog converter was chosen for cost reasons. However, almost any state of the art current output converter can be used. Note that a multiplying converter can provide for scaling of the output by a voltage, possibly from another interface. The LM318 operational amplifier was chosen to be the current to voltage converter to minimize response time of the DAC. As such, only an operational amplifier of similar speed should be substituted unless you can tolerate slower response. Also care must be taken to isolate the amplifier from stray signals, or it could become unstable and oscillate. The multiplexer chosen for both converters is one of the CMOS variety. In order to meet the specifications on this device and give an easy design, the voltages of this system are restricted to ± 5 V. This allows for adequate range for most 8 bit applications. 10 V full scale means one bit is 39.1 mV. This can be scaled down if needed.

The sample and hold capacitor was chosen to satisfy the acquisition and hold time requirements. Changing its value can tailor the system to individual needs. Always use polystyrene capacitors since their characteristics are essential to a good interface. I used the new CMOS operational amplifiers for the buffers because of their high impedance and low cost. Notice that they operate on the same power supplies as the analog switches.

The comparator was chosen for its low cost and speed. Similar devices could be substituted. The necessary power supplies are ±15 V and ±5 V. The ±15 V could be reduced to ±12 V if convenient.

The software for such an interface is not difficult. Roger Frank's article and figure 4 contain the basics. A complete routine written for an MOS Technology 6502 based system is shown in listing 1.

Summary

A multichannel analog interface can be designed with a minimum of hardware if a few software vs hardware trade offs are made. Though the multiplexed approach does impose some software burdens, for most applications the variation of the outputs and inputs is slow enough to make this type of interface transparent under the interrupt system. This type of interface should make many real world applications possible to the limited budgets of most experimenters.■

Listing 1: Typical program written for a 6502 based system to update eight digital to analog conversion channels. The program sequentially addresses each channel, outputs the desired voltage to be held, disables the channel, and steps to the next channel. This is done once for each time that the program is called. This program could be set up as an interrupt handler which responds to a clock strobing an interrupt line.

Label	Op	Operand	Commentary
UPDATE	LDX	#07	initialize pointer;
LOOP	LDA,zpa,X	BUF	get next byte for output;
	STA	DAC	output byte;
	TXA		accumulator:=pointer;
	STA	CONT	select channel and enable sample and hold;
	ORA	04	turn sample and hold strobe off;
	STA	CONT	turn selected sample and hold off;
	DEX		pointer:=pointer–1;
	BPL	LOOP	if pointer >=0 then go to LOOP;
	RTS		else return from subroutine;

Data Definitions

BUF: A string of output bytes
DAC: Address of DAC output port
CONT: Address of control port

BIBLIOGRAPHY

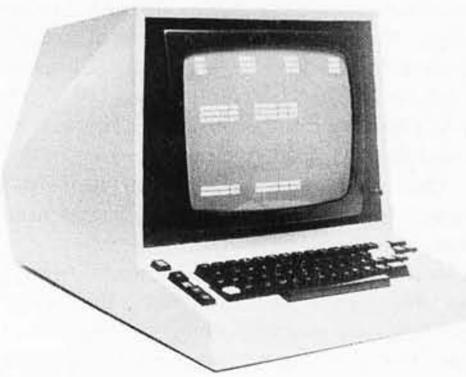
1. Frank, "Microprocessor Based Analog/Digital Conversion," *BYTE* magazine, May 1976, pages 70 to 73.

2. Graeme, Huelsman, Tobey, *Operation Amplifiers: Design and Application,* McGraw-Hill, 1971.

3. Kraul, "Analog Interfaces for Microprocessor Systems," *Electronotes, Newsletter of the Musical Engineering Group*, volume 8, number 63, March 1976, pages 11 to 14.

What's New?

A High Performance Character Display Terminal

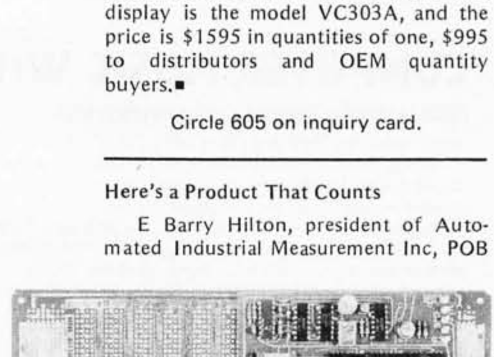


Volker-Craig Ltd, 266 Marsland Dr, Waterloo, Ontario CANADA N2J 3Z1, has introduced a new data terminal intended for use in small business systems, timesharing and other small computer end user applications. The features of this terminal include RS232C interface at rates from 110 to 9600 bps, a 1920 character display (24 lines of 80 characters), 12 inch CRT display, composite video output for extra slave monitors, XY cursor addressing by computer or operator, 64 key ASCII keyboard with tactile feedback and automatic repeat, and an optional separate numeric data entry keypad. Options include upper and lower case ASCII keyboard and display, switched serial interface, parallel input and output interfaces, and custom keyboard character fonts such as APL or French. The display is the model VC303A, and the price is $1595 in quantities of one, $995 to distributors and OEM quantity buyers.■

Circle 605 on inquiry card.

Here's a Product That Counts

E Barry Hilton, president of Automated Industrial Measurement Inc, POB



125, Wayland MA 01778, stopped by BYTE's offices in early February, toting an Altair 8800 under his arm and an interesting peripheral which his firm has designed and is marketing. The peripheral is the AIM-1005 frequency meter, demonstrated to us mounted upon a Vector prototyping card for the Altair bus. (This was a demonstration setup; his firm will soon have a $30 card with all the connections to the Altair bus made for piggyback mounting of the frequency meter on a permanent basis.) What this $178 product does for its user is provide a programmable frequency meter with 13 bits of precision, and 11 different time base ranges allowing measurement over time periods from 10 μs to one hour. The input logic has scaling counters with an upper limit of 25 MHz, so it is possible to make a quite useful general laboratory frequency meter with outputs on a computer terminal by simply driving this peripheral with a simple assembly language or BASIC program. The device is interfaced through memory address space, decoding the high order four bits of the 8080's 16 bit address bus.■

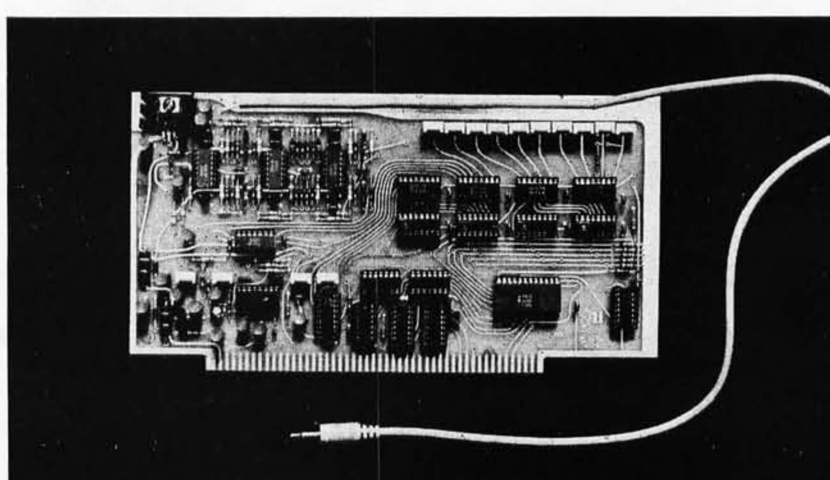
Circle 606 on inquiry card.

A Synthesizer Example . . . .

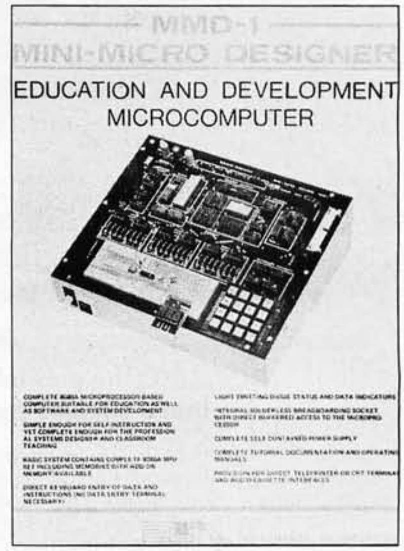
Al Cybernetic Systems, POB 4691, University Park NM 88003, has sent along some new information on their Model 1000 Speech Synthesizer, which was first described in Wirt Atmar's article in August 1976 BYTE, page 26. The picture here is the production version of this device, which is an analog model of the human vocal tract, digitally programmed with commands that correspond to American English phonemes. Since the device works with coded phonemes, the maximum information transfer rate for speech which is required is about 50 bps. The device plugs directly into the Altair bus and cost is $325, with delivery from stock.

A significant and interesting bit of literature which came with this picture was "Programming Example 1," an Altair BASIC program with an 8080 machine language subroutine for details at a low level. The program is an implementation of the well-known Lunar Lander game, but with the novel twist that the "ship's computer" tells you your present height from the surface of the planet as you land. It will run on any Altair 8800 equipped with 12 K of memory. So, what you do is plug in the Model 1000, read in BASIC, type in this program and set up its machine language support routine, then proceed to use a new mode of interaction with the computer system.■

Circle 608 on inquiry card.



A New Brochure



E & L Instruments, 61 First St, Derby CT 06418, has just released a new 8 page brochure describing the MMD-1 education and development microcomputer and its optional accessories. Copies of the MMD-1 brochure are available from the company, as well as its representatives and authorized dealers.■

Circle 607 on inquiry card.

XIMEDIA PRESENTS

The SOROC IQ120

CURSOR CONTROL. Forespace, backspace, up, down, new line, return, home, tab, PLUS ABSOLUTE CURSOR ADDRESSING.

TRANSMISSION MODES. Conversation (half and full Duplex) PLUS BLOCK MODE – transmit a page at a time.

FIELD PROTECTION. Any part of the display can be "protected" to prevent overtyping. Protected fields are displayed at reduced intensity.

EDITING. Clear screen, typeover, absolute cursor addressing, erase to end of page, erase to end of line, erase to end of field.

DISPLAY FORMAT. 24 lines by 80 characters (1,920 characters).

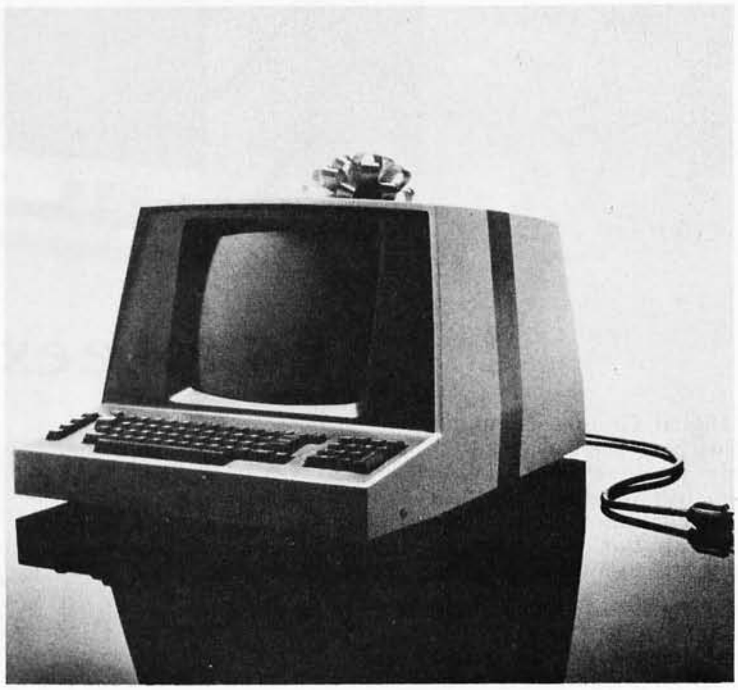
CHARACTER SET. 96 characters total. Upper and lower case ASCII.

KEYBOARD. 73 keys including numeric key pad.

REPEAT KEY. 15 cps repeat action.

DATA RATES. Thumbwheel selectable from 75 to 19,200 baud.

SCREEN. 12 inch rectangular CRT – P4 phosphor.



SPECIAL INTRODUCTORY PRICING

Kit $ 995.00 **Assembled $ 1,295.00**

(Price includes block mode, lower case and 24 line options.)

Specials of the Month

North Star **Micro Disk** with power supply and cabinet	Kit – $699
	Assembled – $799
IMSAI I-8080 with **TDL** ZPU	Kit – $825
	Assembled – $975
Digital Systems FDS Disk Drive with **CP/M** Software (assembled only)	Single – $1,750
	Dual – $2,350
Mountain Hardware **PROROM**	Kit – $145
	Assembled – $210
Vector Graphic 8K RAM	Kit – $235
	Assembled – $285

NOW WE'RE TOLL FREE

800-227-4440

(in California, Hawaii, and Alaska, call collect: 415/566-7472)

XIMEDIA OFFERS A FULL RANGE OF PRODUCTS FOR THE PERSONAL COMPUTER ENTHUSIAST AND THE SMALL SYSTEM DESIGNER. LET US QUOTE ON ALL YOUR HARDWARE AND SOFTWARE NEEDS.

OUR RETAIL STORE – THE COMPUTERIST™ – IS NOW OPEN IN SAN FRANCISCO. CALL US FOR DIRECTIONS.

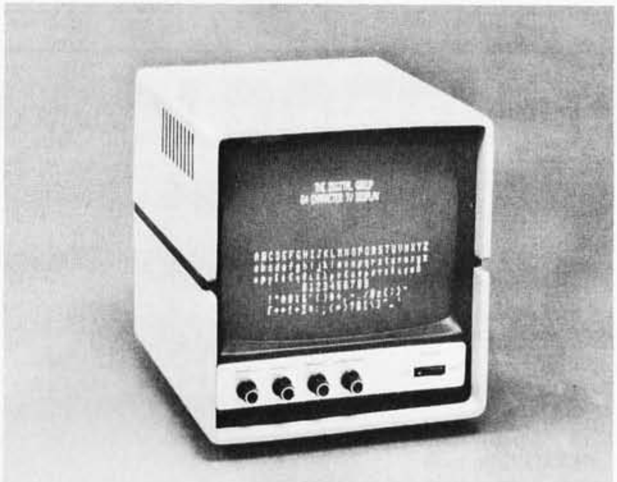
XIMEDIA

1290 24th Avenue, San Francisco, CA 94122
(415) 566-7472

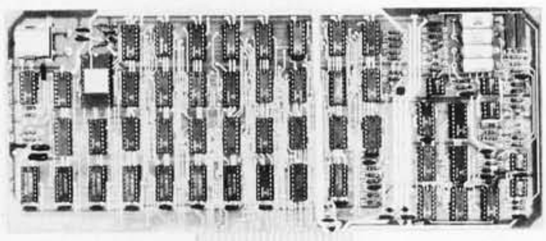
Circle 220 on inquiry card.

COD orders freight collect. Orders with payment shipped prepaid. California residents add 6% sales tax. Please allow 3 weeks for delivery.

The Digital Group adds character(s).



64, to be exact.



The Digital Group's computer systems have a lot of character already. Just one quick look at any of our products in their unique custom cabinets confirms that. But we believe it never hurts to add a bit more.

So, the Digital Group has added character in a big way to give an added dimension to the operation of our video-based computer systems. We are pleased to announce our new TV readout with a 64-character line. It will give your system a great deal more capability. Give it more character, if you will.

Here are the specifics on the Digital Group TV Readout and Audio Cassette Interface:

1024 Character TV Readout

- 64 characters horizontal by 16 lines
- 7x9 character matrix (effectively 7x12 due to character shifting)
- 1K on-board RAM for buffer storage—requires no main memory—completely independent
- 128 character ASCII
  - Upper case alpha
  - Lower case alpha with base line extenders (g, j, p, y)
  - Numbers and extended math symbols
  - Greek alphabet
- Software driven cursor—forward and backward
- Compatible with most microprocessors; Interfaces with 1 8-bit parallel output port
- Timebase may be driven with an external timebase (may be synchronized to TV camera, TV set, etc.)
- Readout timebase available at connector (can be used for graphic driver, etc.)
- White characters on black, and/or black on white; software selectable
- Plugs into standard dual 22-pin TVC connector on Digital Group Systems

Improved Audio Cassette Interface:

- Reliable FSK recording technique
- Uses standard unmodified audio cassette recorder
- Write cassette system uses a digitally synthesized frequency shift system, derived from TV system's master crystal oscillator
- Read cassette system easily aligned using the write system as an alignment aid.
- Runs at 1100 baud (100 characters/second)—loads 16K in 3 minutes

512 TVC to 1024 TVC Upgrade Kit:

As always, when the Digital Group extends the capabilities of our systems, it doesn't mean obsolescence for any products. We are offering an upgrade kit for present Digital Group system owners who wish to go to the longer line length. This kit uses most of the IC's from our TVC-F readout. No unsoldering is required; all new sockets, capacitors, resistors, PC board and other necessary parts are supplied.

Prices:

TVC-64—Full 64-character TV Readout & Audio Cassette Interface:

Kit — $140 Assembled — $205

TVC-64UPG—Upgrade kit from TVC-F:

Kit — $65

If you already own a Digital Group system, our 64-character line will definitely enhance its operation. If you're just looking, you might want to keep in mind that the Digital Group has a lot of characters.

Write or call now for details on our new 64-character TV readout and all our other exciting products.

the digital group INC

box 6528 denver, colorado 80206 (303) 777-7133

Newt:

A Mobile, Cognitive Robot

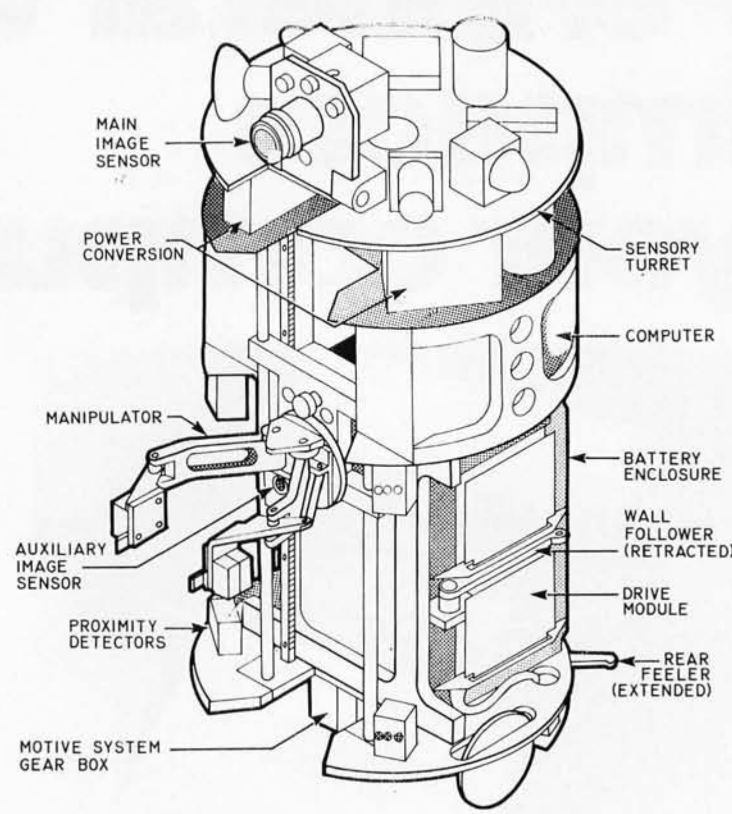


Figure 1: The robot Newt as it will appear when completed. There are three main subsystems shown in this diagram which was supplied by the author: motive, manipulator and sensory turret. The entire machine is controlled by an on board microcomputer. The motive power subsystem is built around two precision drive wheels actuated by stepper motors. The manipulator, a simple hand capable of grasping, lifting and rotating, is also actuated by stepper motors and includes various sensors. The sensory turret at the top is a platform which includes a main image sensor, which can be tilted to look straight down or up at a slight angle using a stepping motor; other sensors are shown in outline form, and are a subject for future experimentation.

Ralph Hollis
Dept of Physics and Astrophysics
University of Colorado
Boulder CO 80309

In the late 1930s, a young man named Rossum began manufacturing industrial robots in a small factory on the outskirts of Prague. This venture was immediately successful and would have virtually guaranteed a second industrial revolution had it not been for a singular tragic circumstance: The robot workers became irrational and revolted. They turned on their masters and burned the factory to the ground.

Fortunately, the preceding scenario is only a work of fiction by the Czechoslovakian writer K Čapek. Since Čapek's coinage of the word "robot" in his 1923 play R U R it has been the subject of a great many works of science fiction, including Isaac Asimov's *I, Robot* and the movies "Forbidden Planet," "Gog," "Silent Running," "Westworld" and others.

Alas, the imaginative stories of the science fiction writers have far outstripped the efforts of the robot engineers. Progress in building real robot devices has been painfully slow over the past several decades, although a few individuals and small groups have produced some very interesting results.

It is not my purpose here to review these works, but rather to describe a project with which I have been personally involved for some time: the construction of a freely moving robot vehicle which will be capable of interacting with its environment in a rational way, and managing its own survival. The work is being done by a small group of people in the Duane Physical Laboratory of the University of Colorado. The project is being financed through personal funds, and in this sense qualifies as an "amateur" undertaking.

What are the requirements for such a robot? It must be able to explore the environment in some orderly manner, measure the attributes of objects and obstacles encountered, classify them according to some scheme, and incorporate them in its evolving internal world model. The world model must have a logical structure which allows modifications to be easily made; it must be compact with respect to memory space, and it must use a design which can be consulted in some reasonable way. The roLot must be able to manipulate the world model (cognition), derive informed decisions from it, and carry out these decisions in physical action to achieve broadly defined goals.

Figure 1 is a general view of the robot as it will appear when completed. Practically speaking, it resembles less a mechanical man than, say, a shop type vacuum cleaner. The machine is cylindrical, about 36 cm (14 inches) in diameter and 76 cm (30 inches) in height, weighing approximately 27 kg (60 pounds). All rigid mechanisms lie within the cylindrical boundary or can be retracted within the boundary if necessary. This design greatly reduces the number and diversity of senses required. The robot will be fitted with a smooth cylindrical skin (not shown), removable in sections for easy inspection and maintenance. Modular construction is used in the robot wherever possible. There are three main subsystems: motive, manipulator and sensory turret. These are presided over by an 8080 based microcomputer with 8 K bytes of EROM and 24 K bytes of programmable memory. The entire system is powered by a 6 V, 84 amp-hour storage battery.

How the Robot Gets Around

The robot is given locomotion by two main diametrically opposed drive wheels. A third (unpowered) castoring wheel is located at the rear. This motive geometry has been successfully employed in several other robot vehicles constructed by other groups, and has been in use in the author's robot research since 1957. The vehicle's center of gravity is located well aft due to the placement of the battery and other heavy components, obviating the need for a front wheel. The wheels are constructed of aluminum with neopreme O-ring tires. The main wheels are driven by stepping motors through precision 3:1 gearing. A single command step rotates a wheel 0.6 degrees, so there is a total of 600 steps per revolution. If both wheels are moved in the same direction, the vehicle travels forwards or backwards in a straight line. If the wheels are moved in opposite directions, the vehicle executes a perfect rotation about its vertical axis. By stepping the wheels at different rates, a circular trajectory is approximated having a radius which can range from zero to infinity. The hardware and software necessary to drive the stepping motors are discussed in some detail later in this article, since they have very general application.

The robot navigates principally by open loop dead reckoning. That is, it depends largely on the precise control of the wheels during acceleration, deceleration, and constant speed motion to achieve accurate positioning of the vehicle. The robot's position and azimuth relative to a fixed origin are computed at the end of each motion segment by counting stepping motor increments and using trigonometry. The precision attainable is limited principally by wheel slippage, unequal wheel diameters, nonplanar floors, round-off errors, and step quantization. In practice, all these errors are small for short distance movements.

The robot's excellent open loop performance makes it unnecessary to have continuous closed loop servo systems, such as have been extensively employed in other robots. For example, consider the problem of having a robot view a small object on the floor a short distance away, and then go pick up the object. One approach would be to have the robot continuously view the object as it moves towards it, adjusting its motion in a continuous way to converge on the object and pick it up. In this approach, a rather crude motive system would suffice, since errors are always nulled. A heavy load, however, is placed on the sensor-computer system. An alternative approach, the one followed here, is to have the robot view the object once, and then compute exactly how to move to the object and pick it up. When the computation is completed, the machine simply carries out the proper motions in a blind fashion. There is a greatly reduced strain on the sensor-computer system, at the expense of having to build rather precise motive machinery. Of course, this approach assumes the relevant environment will remain fixed for the duration of the task.

In this general spirit, by using high precision open loop movements throughout the robot, only intermittent feedback through the senses is required to close the control loop. In this way, the overall complexity of the robot can be kept at a reasonable level, allowing the required computations to fall within the abilities of an on board microcomputer.

Epistemological Engineering

With artificial intelligence, robotics and applied cybernetics coming of age through the recent progress in the fabrication of computer systems, it is quite likely that the branch of philosophy called epistemology will have a much more explicit role in technology during the coming years. Epistemology is the study of nature and grounds of knowledge; understanding of epistemology is crucial to any attempt to realistically implement artificial intelligence systems. So in a future world of cognitive automata and advanced information systems, we may indeed find the new specialist who is the "epistemological engineer."

Scanning the Environment with Electronic Senses

The sensory turret (see figure 1) provides a general platform on which to mount various senses, some of which might be quite experimental and temporary in nature. It can pan a full 360 degrees, and a small section containing the main image sensor can tilt from approximately 30 degrees above horizontal to 90 degrees below horizontal. Both of these motions are controlled by stepping motors. The main image sensor has a motorized focus control. The geometry of the "hand-eye" system allows orthogonal views of objects held in the manipulator jaws by using both image sensors under conditions of controlled focus and lighting.

Each image sensor is a 32 by 32 element integrated array of photodiodes on a silicon chip measuring approximately 4 mm (0.1 inch) square. The array acts like a 1024 bit memory. Each element is precharged to a fixed voltage; then at some later time the voltage of each element is read out, decreased in proportion to the amount of light which has fallen on it. Several milliseconds are required to digitize the image. Up to 16 levels of gray can be discerned, which means that 512 bytes of computer memory are sufficient for a single image. Once the image is obtained, it is analyzed by appropriate software. The extremely small size of these solid state image sensors, and the simplicity of their associated electronics, make them ideal for robot use. Higher resolution devices such as charge coupled sensors and miniature television cameras are available, but their cost and complexity make them unattractive for such applications. Besides, the amount of data generated would be too unwieldy to handle with a microcomputer.

In addition to the main image sensor, several simple phototransistor light sensors are mounted on the tilting platform of the sensory turret. These enable the robot to locate and track point sources of light.

Several other proposed senses are to be mounted on the turret. These are shown in schematic form in figure 1. To the left of the image sensor is shown an ultrasonic ranging transmitter and receiver which should be extremely useful for finding the range to walls at distances as great as 10 meters (33 feet). The idea is to transmit short bursts of 40 kHz sound and measure the time required for an echo to be received in order to compute the distance. Much of the necessary circuitry is available in compact integrated form.

Just to the right of the image sensor is shown a long wavelength infrared radiation detector with which it might be possible to locate sources of heat such as dogs, cats and humans.

Located on an axis perpendicular to the main image sensor are shown two microphones which form part of a sound location system. The intent is for each sensor to acquire a short sample of sound. The computer then attempts to find a phase relationship between them, thereby locating the direction of the source.

Many other possible senses can be imagined; the sensory turret is intended to provide a versatile platform and interface for experimenting with them. For example, using a helium neon laser, or perhaps a compact metal vapor laser, it may be possible to provide an optical range finding system with high resolution if a suitable detector can be found. Eventually it is hoped to provide limited forms of speech synthesis and phrase recognition (using off line electronics with an analog radio link). This would make it possible to give general spoken commands to the robot in contextual surroundings, and have verbal feedback to insure that the commands were being properly interpreted.

Sharing some space with the sensory turret system, and directly below the platform in figure 1, is the necessary power conversion electronics. These modules convert the battery voltage to ±12 V and +5 V (regulated) for use by the electronics, and also house the battery recharging circuitry.

Manipulating Objects in the Environment

The number of tasks which a simple robot can do is greatly increased if it has some sort of hand with which to grasp and manipulate objects in the environment. The human arm and hand system, with its 27 degrees of freedom, is a marvelously versatile mechanism. Large industrial manipulators are fairly complicated and have six or seven degrees of freedom. For practical reasons, the present system is limited to a mere three degrees of freedom which, when combined with the two degrees of freedom of the motive system, should be sufficient for carrying out simple tasks.

The manipulator is able to grasp objects, move up and down along the front of the robot by means of a rack and pinion drive, and to rotate about a horizontal axis. All three motions are controlled by stepping motors. The manipulator has a parallelogram geometry which permits the jaw faces to remain parallel regardless of their separation. This feature simplifies the problem of picking up objects of varying sizes. When not in use, the jaws open wide enough to bring all parts of the manipulator within the cylindrical boundary of the robot, where it is out of the way and does not cause problems when the robot turns and maneuvers in tight places.

An auxiliary 32 by 32 element image sensor with fixed focus lens is mounted directly in the manipulator assembly, providing a view of whatever object is between the jaws.

In addition to mechanical force sensors in the jaw faces, there are several infrared LEDs in one face with opposing phototransistors in the other face. These LED phototransistor pairs define beams of light between the two jaw faces. By scanning the manipulator up and down, and moving the entire vehicle forwards and backwards, these beams enable the robot to measure the height and depth of simple objects resting on the floor between the jaw faces, as well as to determine when objects are correctly positioned for picking up. The widths of objects are measured directly by counting the number of stepping motor pulses required to close the jaws on the object.

A tactile sensor is mounted on the front of each jaw to enable the robot to sense and locate large obstacles such as walls. The robot can align itself accurately either parallel or perpendicular to the wall, after performing a simple series of maneuvers. This action is convenient for many of the tasks the robot might have to carry out. For example, a simple strategy of maneuvers allows the robot to "square itself off" from a corner in a room, thereby defining a precise origin for its navigational coordinates.

The tactile sensors double as electrical contacts used for plugging into ordinary wall outlets for the purpose of recharging the robot's battery (more about this later).

All electrical power and signals to and from the manipulator assembly are sent through folding ribbon cables which are not shown in figure 1.

Auxiliary Senses

In addition to the senses mounted on the manipulator and sensory turret assemblies, there are several auxiliary senses illustrated in figure 1. A pair of retractable feelers are provided in the rear of the vehicle and serve much the same purpose as the tactile sensors on the manipulator jaws. These rear feelers enable the robot to detect and square up from walls and corners if the manipulator is occupied with holding some object.

In addition to the rear feelers, a pair of retractable wall-follower feelers located on the sides of the vehicle enable the robot to travel parallel to a wall by measuring the distance from itself to the wall. The rear feelers and the wall-follower feelers are both activated by double coil latching solenoids and contain "microswitches" consisting of movable masks and infrared LED-phototransistor pairs.

For all of the robot's varied and specialized senses we have discussed so far, there is nothing to prevent the robot from accidentally running at high speed into a wall. To prevent just such a mishap, and to provide a "soft" broad area "sense of touch," the robot is equipped with a number of proximity detectors, several of which are shown in schematic form in figure 1. These detectors work on a simple, but elegant, principle. Infrared LEDs, amplitude modulated by a 20 kHz signal, send broad beams of light out from the robot. If a wall is nearby, some of this light is reflected back into phototransistor sensors located in the same module. The received signal is amplified and sent to a phase sensitive detector locked to the outgoing signal. The computer is notified if the signal exceeds some programmable threshold. These proximity detectors are quite insensitive to ambient light conditions, not very sensitive to the color or texture of the reflecting surface, and have useful ranges extending from several centimeters to perhaps one meter. With these detectors, the robot is free to travel at fairly high speed until a wall or other obstacle is encountered, and can then slow down and investigate it with caution.

It should be mentioned that all of the senses interact with the processor through a vectored priority interrupt system.

Robot Psychology

Up to the present time, the environment of the robot has been intentionally restricted. The robot is permitted to roam freely within a large "playpen" constructed of plywood. As various sensory functions are added, and the robot nears completion, simple objects such as blocks of wood, and larger obstacles constructed from plywood will be introduced, and restrictions on the environment will be gradually relaxed. While this is happening, the complexity of the software will greatly increase. It is expected to take several more years of effort before the robot reaches a form which can be considered "finished." Whether the robot will ever be able to cope successfully with a "general" environment, such as a typical research laboratory or home living room, is certainly an open question.

The software will be developed as a hierarchy of modules. The bottom strata of

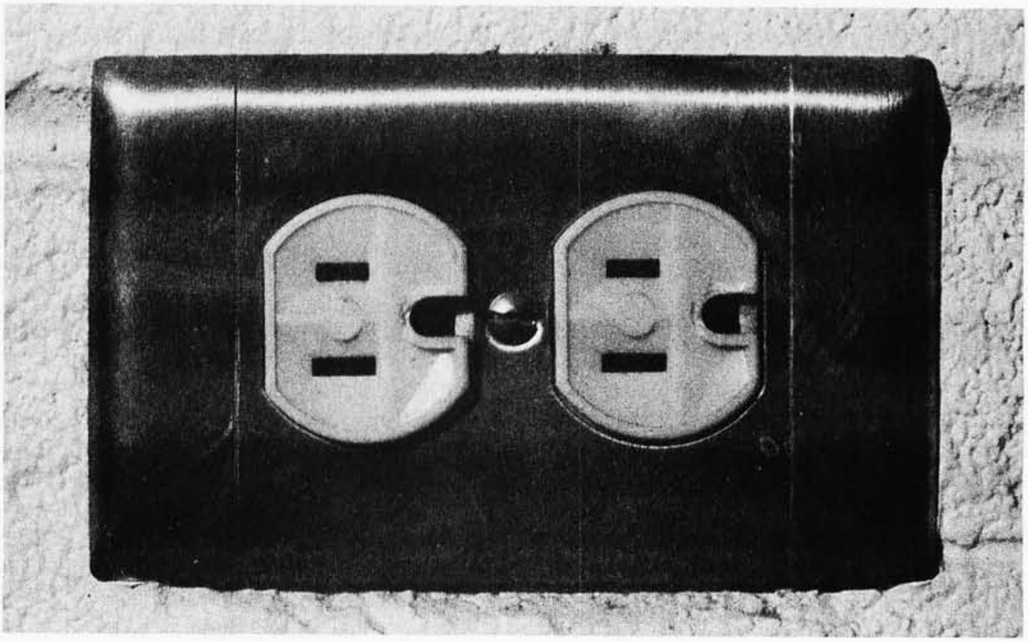


Photo 1: The feeding trough. This is a photograph of an ordinary wall outlet with a white concrete block wall as a background. The robot must be able to seek out and plug into outlets such as this in order to periodically replenish its energy supply.

this hierarchy will be occupied by hardware oriented routines for motor control, interrupt servicing, sensory data acquisition and the like. These routines will be relegated to EROM, where they will reside as powerful appendages for the higher level software. In the highest reaches of the hierarchy will be the planning algorithms: heuristic programming which will take into account goals, subgoals, behavior, the structure of priorities, and what may generally be called the "psychology" of the robot.

If the robot is turned loose in a strange room, its immediate behavior will be to begin exploring the room in some systematic way, locating walls and the boundaries of objects with respect to a specific origin. As new features of the environment are encountered, they will be incorporated into the world model. Particular interest will be shown in the positions and heights of recognizable wall outlets, since these are critical for survival. As the model of the room nears completion, the original intense curiosity will subside, and will be replaced by an attitude of playfulness. The world model will be consulted to find out which objects (eg: wooden blocks) are small enough to be manipulated. The robot may choose to group together objects with similar characteristics, build simple structures by stacking blocks and so forth. When the energy supply nears exhaustion, the robot will interrupt its play, head directly for a convenient wall outlet, plug in, and recharge its battery. The recharging process might take several hours, during which all motors will be turned off, and all but one 4 K byte memory module will be powered down. After recharging, the robot will unplug itself and continue on its way. When the robot eventually "tires" of its environment, it will perhaps leave the room, wander down the hallway, and look for other rooms to explore. An interesting experiment would be to observe how long it takes for the machine to recognize a room in which it is arbitrarily placed, but has previously explored. In addition to observing general behavior, one can give the robot general tasks to perform, such as picking up all small objects in a cluttered room and placing them in a box in the corner. (Mothers with small children, take note!)

Many of the computing requirements for the robot will exceed the capabilities of its on board microcomputer. For example, the analysis of complicated scenes viewed by the image sensors can, at present, be done only by a very fast computer with lots of memory. For this reason it is planned to equip the robot with a duplex radio telemetry link to a "black box" which connects with a telephone. The robot will then be able to initiate telephone calls to a large timeshared computer, and be able to communicate with it via the 2400 bps telephone lines. Control routines in the on board microcomputer will invoke large analysis programs stored on permanent disk files attached to the timeshared computer.

A visual scene can be transmitted in less than a second, and the large computer can extract features of interest and send them back to the robot in several seconds. It will also be useful to take advantage of the off line disk files to store large data bases such as portions of the world model. When the robot has finished its transactions with the large computer, it simply "hangs up the phone" and goes about its business, perhaps reinitiating the hookup at a later time. It is important to point out here that in no sense is the large computer "in control" of the robot. The robot is simply using the services of the large computer to perform calculations that are too involved or too lengthy to do itself, much as you or I would use a computer to solve some problem. Also, it should be noted that if the large timeshared computer is busy, or if the computations are long, the robot is free to continue with other tasks until the requested analysis can be performed.

This usage of an off line timeshared computer will permit access to a large amount of general robot planning software written in high level languages by other groups. The telemetry hookup will, as a side benefit, enable the robot to make calls to human researchers (hopefully not in the middle of the night) to report malfunctions or unusual conditions in the environment by means of coded audio signals. (The robot night watchman?)

Finding Energy Sources

It may be necessary for the robot to survive for weeks at a time without supervision or interruption of its operation. To do so, it must manage its own energy supply. The battery voltage and current are sensed periodically and converted to digital form by a simple software driven analog to digital conversion system. This information, along with the battery's internal temperature (sensed with a thermistor probe), allows the state of charge to be determined. When this reaches some minimum level, the robot is obliged to renew its supply of energy by finding a wall outlet and plugging in.

From the outset, it must be said that the robot will depend heavily on preprogrammed strategies to accomplish this task. A standard wall outlet mounted on a white concerete block wall is shown in photo 1. The cover plate is of brushed stainless steel, and the receptacles are made of light colored molded plastic. From a distance of perhaps 10 meters (33 feet), with the turret image sensor focused on infinity, the outlet appears as a small spot. Any such spot, identifiable by the software as a few dark picture elements surrounded by a light field, is worth further investigation. If several spots are visible, a single one will be selected for closer examination.

Using the motive system and the sensory turret in combination, the direction and distance to the spot is then computed by triangulation. After this is done, the robot

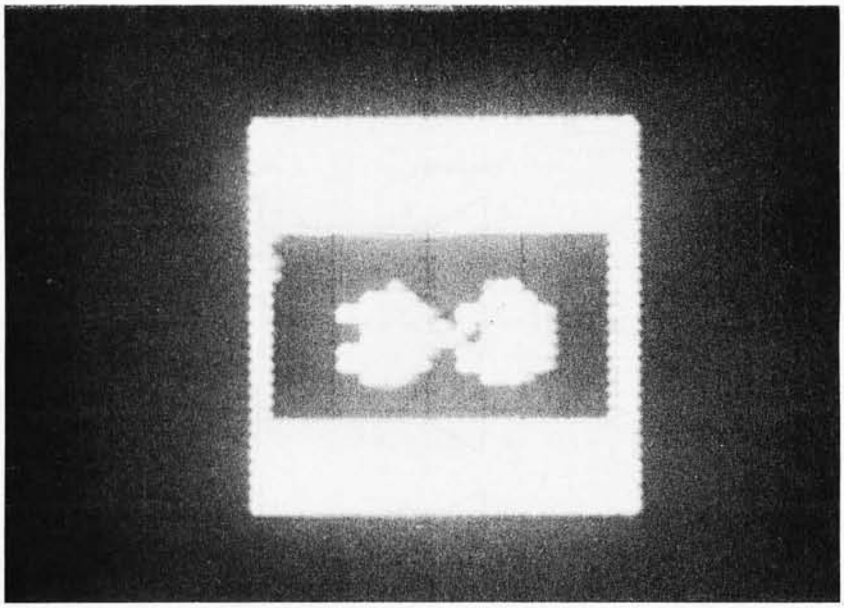


Photo 2: Here is an oscilloscope display of the wall outlet shown in photo 1, as viewed by an image sensor. There are a total of 1024 picture elements in this sensor, each with 16 levels of gray perceptible to the computer. The picture has 32 columns and 32 rows of elements. The outlet appears as a dark rectangle with a light background. The two receptacles can just be resolved at this range.

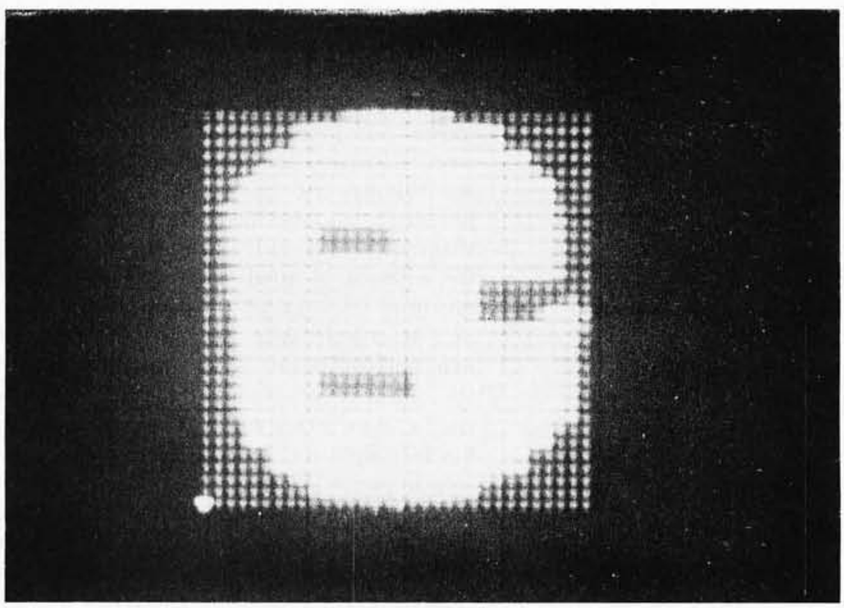


Photo 3: As the robot moves closer to the wall outlet, more detailed features of the socket become apparent. This is an oscilloscope display of an individual receptacle, as viewed at close range by the image sensor. By applying pattern recognition techniques to images such as this and the image in photo 2, the robot can recognize outlets and determine the exact position and height of the electrical contacts, prior to its final maneuver to plug itself into the wall.

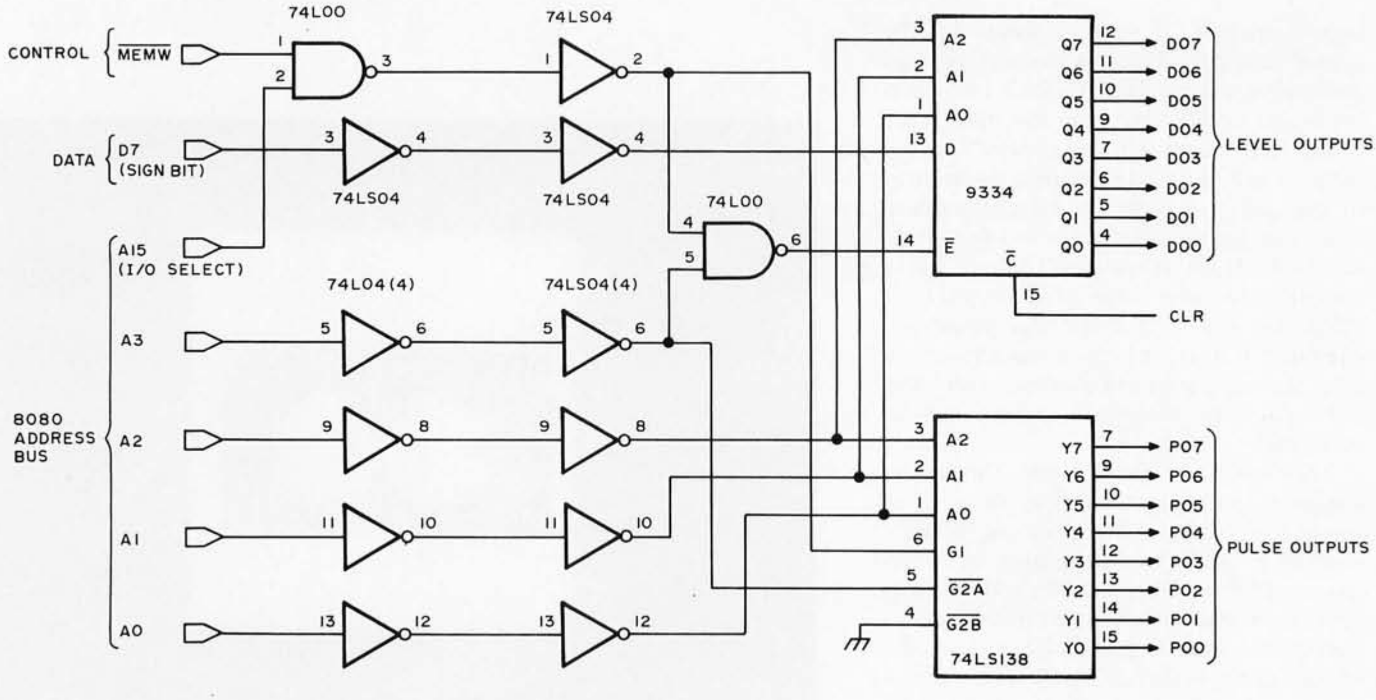


Figure 2: Output decoding logic used to create discrete (level) outputs and pulse outputs. This circuitry connects to an 8080 data bus through memory address space. The outputs are used to drive the various lines of stepper motor interfaces such as those shown in figure 3. Each stepper motor individually uses one discrete level output (for forward or reverse direction control) and one pulse output (to initiate one step sequence). Two discrete level outputs are also used to control the running versus standby, or off, status of the motors.

drives to the approximate position of the spot, and with LED proximity sensors and the tactile sensors on the manipulator, it attempts to sense and recognize the existence of a wall, which is hopefully associated with the spot. If this is the case, the robot moves out perpendicular to the wall for a fixed distance, sets the image sensor focus accordingly, and rescans for the spot which hopefully should now begin to have the shape of a wall outlet from this short range. By a series of small maneuvers, the robot positions itself exactly in front of the outlet at a precise distance away. At this time, the manipulator begins a vertical motion until the outlet is centered in the field of view of the auxiliary image sensor. The computer now has a straight on, in focus view such as that seen in photo 2.

At this point, fairly simple pattern recognizing algorithms are used to analyze the shape, size and topology of the image (feature extraction). These features are compared with known properties of wall outlets. If enough features match within certain error bounds, the recognition is successful. (In this respect, it is much easier to attempt to recognize a specific object than to try to analyze a general scene.) If the recognition is not successful, several more attempts are made at slightly different distances to overcome the granularity effects of the sensor. If a recognition is still not successful, the robot heads off to explore other spots. Assuming success at this point, the robot moves in for a high resolution image of a single receptacle (see photo 3). Again, the image is analyzed, and further recognition tests are made. If everything is okay, the exact height and distance to the electrical contacts are determined, and the manipulator jaws close to the separation appropriate to the space between the two contacts. An attempt is now made to plug in, with the computer monitoring the voltage between the two plug prongs on the manipulator. Small searching motions are made until contact is established, whereupon the jaws open slightly to make good electrical connections. The computer then directs the recharging operation until the correct charge is reached. It is expected that the robot will normally be able to go about six hours between recharging operations.

Stepping Motor Drives

In order to execute all of the motions required of it, the robot must have precise control of the motors. This is accomplished by means of the stepping motor drive modules which constitute the interface between the on board computer and the stepping motors. There is a total of seven identical drives: two for the motive system (one for each wheel), three for the manipulator (grasp, lift and rotate), and two for the sensory turret (pan and tilt).

Figure 2 illustrates the logic used to provide both pulse and discrete (level) outputs from the computer. For simplicity, only eight pulse outputs and eight discrete outputs are shown. The scheme can be expanded to many more outputs, with the addition of suitable logic. Low power TTL provides the connection to the address, data and control buses of the microprocessor. Lines A_0 to A_3 select one of 16 possible outputs, with A_{15} acting as an input output select line. The sign bit, D_7, is used to signify whether a one or a zero will be written into the selected discrete output line. Generation of the negative going output pulses and strobing of the data are done by the $\overline{\text{MEMW}}$ pulse. A 9334 addressable latch and 74LS138 binary to octal decoder provide the outputs to the system. If desired, one pulse output can be wired to the CLR line of the 9334 latch for simultaneous resetting of all the discrete outputs.

Figure 3 shows an individual stepping motor drive circuit which is sufficiently simple and general to permit many other non-robot applications where precise computer controlled motion is required. The circuit generates the 4 phase pulse sequence of high currents necessary to operate a Slo-Syn™ bifilar type stepping motor manufactured by the Superior Electric Company. Inputs to the stepping motor circuit are connected as desired to the discrete and pulse outputs shown in figure 2. The forward or reverse input ($F\overline{R}$) determines whether the motor shaft turns clockwise ($F\overline{R}$ = 1) or counterclockwise ($F\overline{R}$ = 0) as viewed from the shaft end of the motor. The forward limit input ($\overline{FL}$) prevents the motor from moving forwards (clockwise) if $\overline{FL}$ = 0. The reverse limit input ($\overline{RL}$) prevents the motor from moving backwards if $\overline{RL}$ = 0. These inputs are generally not controlled by the computer, but are wired to simple limit switches to prevent excessive motion of the motor in a particular direction. In the case of the robot, these "reflex inputs" are independent to allow the computer to change the direction of motion after a mechanical limit has been reached, and to establish "zero point" settings for the various possible motions. The pulse input ($\overline{P}$) triggers on a negative going transition and causes the motor to advance one step (1.8 degrees) in the direction specified by the $\overline{FR}$ input. Thus, 200 pulses on the $\overline{P}$ input cause the motor shaft to rotate exactly one revolution. The maximum pulse rate is of the order of several hundred pulses per second without error, but this depends strongly on the motor size and the driven load. The clear input ($\overline{CLR}$), when momentarily brought low, resets the counting flip flops to zero in case it is necessary to establish a standard shaft position when power is turned on. The off input ($\overline{OFF}$), when brought low, removes all current from the motor windings, and frees the shaft except for a small holding torque caused by the permanent magnets. If $\overline{OFF}$ = 0 and standby (SBY) is brought high (SBY = 1), the shaft position is maintained, but at greatly reduced current. The off and standby inputs are critical for application in the robot, since they permit substantial power savings. In figure 3, series 7400 logic can be substituted for the 74L and 74LS series if desired, but at the expense of extra power. Resistance values in the drive circuits are typical, and might need to be optimized for a particular motor. The diodes in the emitter circuit of the 2N3055s should have a high surge rating, and moderate heat sinking for 2N3055s is advised.

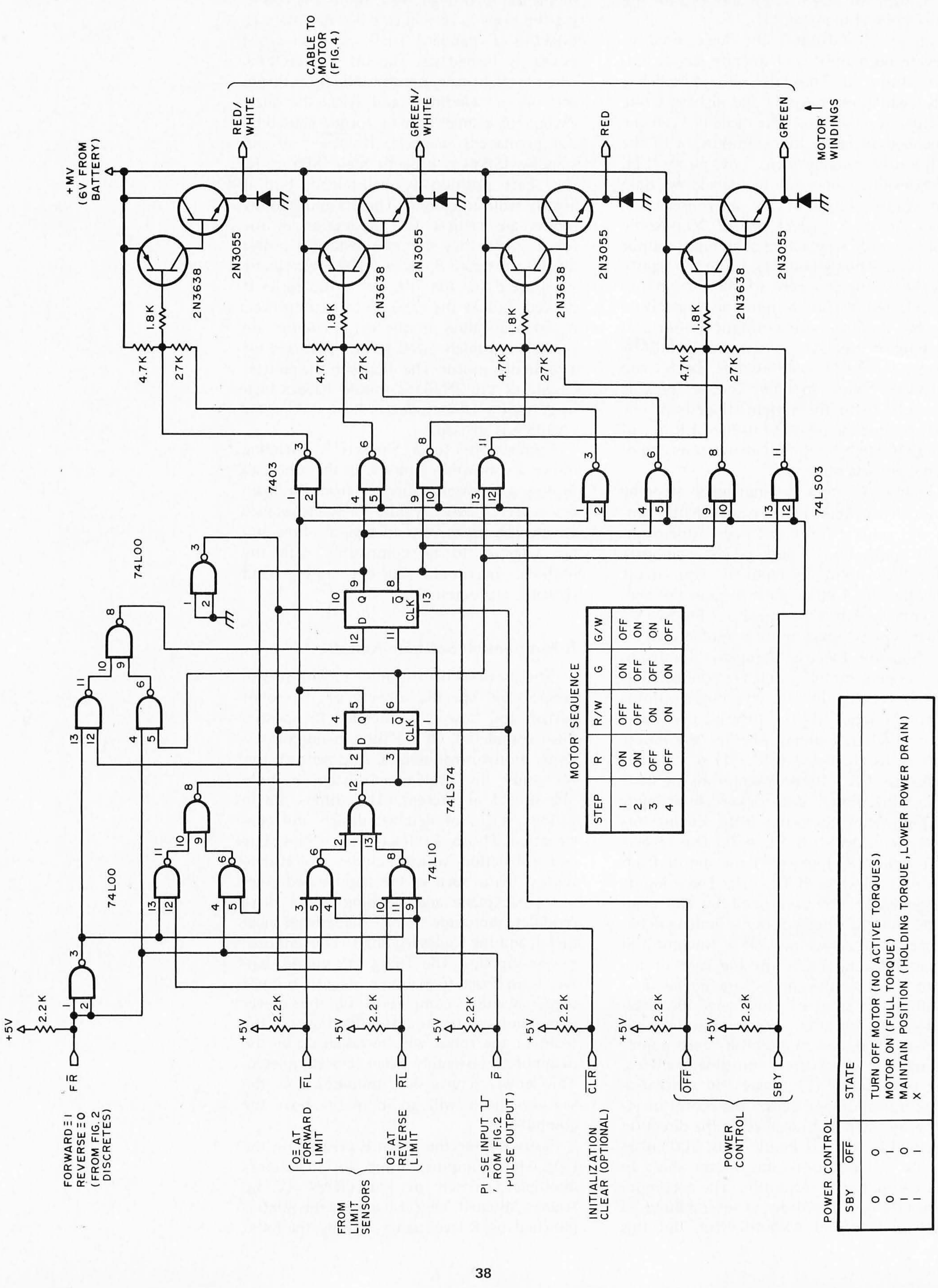
Connections to a Slo-Syn™ stepping motor are shown in figure 4. In the robot, all motors are powered directly from the main 6 V battery (+MV). Values for the resistance R can vary from zero to several ohms and are selected to be compatible with the motor's maximum current rating and dynamic characteristics.

Robot Control: Software Aspects

The previous discussion dealt with several general and specific aspects of the robot system and how it is intended to operate, once completed. Of the three major subsystems, motive, manipulator and sensory turret, only the motive subsystem is fully operational at present. The others are in various stages of detailed design and construction. Photo 4 reflects the current state of construction, which includes several temporary items such as the hand wired computer backplane and stepping motor drive modules, the crude "hand" made from sheet metal, and the cables leading to conventional power supplies. The 16 kg (35 pound) battery is on board to maintain balance, but not used, awaiting completion of the power conversion electronics. The open area at the front of the robot will be taken up by the manipulator assembly when it is completed. The power conversion modules and the sensory turret will go in place above the computer.

Software for the robot is created on the CDC-6400 computer using an 8080 cross assembler written in FORTRAN IV by Robert Mitchell. Object code for the 8080 is punched on 8 level paper tape by the 6400

Figure 3: Each stepper motor (there are seven presently incorporated into Newt's design) requires a drive circuit as shown here. This circuit contains counting logic and high current drivers needed to create the 4 step sequence of signals required by a



MOTOR SEQUENCE

STEP	R	R/W	G	G/W
1	ON	OFF	ON	OFF
2	ON	OFF	OFF	ON
3	OFF	ON	OFF	ON
4	OFF	ON	ON	OFF

POWER CONTROL

SBY	$\overline{OFF}$	STATE
0	0	TURN OFF MOTOR (NO ACTIVE TORQUES)
0	1	MOTOR ON (FULL TORQUE)
1	0	MAINTAIN POSITION (HOLDING TORQUE, LOWER POWER DRAIN)
1	1	X

Slow-Syn TM stepping motor. The inputs to this interface circuit include one level ($F\overline{R}$) signal and one pulse signal ($\overline{P}$) derived from an output port such as the one shown in figure 2. The forward and reverse limit inputs are used to override commands when a sensor detects that rotation of the motor has reached some mechanical limit of the motor's activity. The power control logic (summarized in the chart) is used to conserve battery power by allowing a "standby" mode for the motor when it is not being actively driven. This standby mode holds the motor shaft positon actively, but uses less current than the full torque state used to move the shaft from step to step.

system, and is subsequently read into the robot's memory through an ASR33 Teletype. A modest resident monitor program is stored in the first 666 bytes of EROM, which allows dumping and modifying of memory, punching and reading of paper tape, and branching to any memory location. The monitor is essential, because the computer has no conventional "front panel" with switches and lights.

The current robot control program occupies about 1000 bytes of programmable memory, not including table areas of variable size. It provides a method for exercising direct control of the robot for testing purposes. The coding, written by Dennis Toms, is very general, modular and compact. In order to start and stop the stepping motors without error, it is necessary to provide a profile of acceleration and deceleration. This is most conveniently done by using a time-delay table giving the appropriate time intervals between stepping motor pulses. The table is precomputed to yield a uniform acceleration over some time period. It is sufficient to have a single table for all motors. Each motor has associated with it a 14 byte motor status word (MSW) whose format is given in figure 5. The motor number and motor flag each occupy one byte; the other entries are two bytes (one word) each. The motor number byte is a fixed number which associates the MSW with a particular motor. The motor flag byte is a code which gives the current state of the motor (off, on, standby, accelerating, decelerating, emergency stop, etc). The speed pointer word is an index which points to the current entry in the time delay table. The top speed word specifies the index to the delay table entry, giving the shortest permissible delay between stepping motor pulses (maximum motor speed). The total steps word specifies the number of steps the motor is to execute. The acceleration, constant speed and deceleration counters specify the number of steps to be taken in the acceleration, constant speed and deceleration phases of the motion, respectively. These counters are decremented appropriately as the motion takes place, so that if necessary, at any time during the motion the state of the motor can be determined by

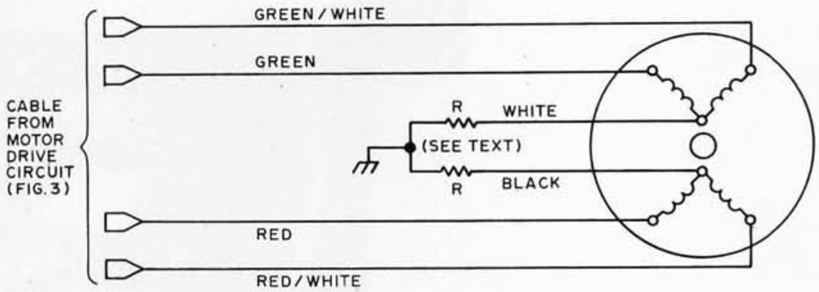


Figure 4: Wiring and color code for the Superior Electric Slow-Syn TM bifilar stepping motors. The resistance R should be set (see text) to reflect typical operating characteristics of the motor's use. The wiring is shown by color designations, and is connected to the drive circuit of figure 3. The ground return to figure 3 can be via the robot's chassis or through an additional circuit in the cabling.

interrupt handling routines. After all bytes of the appropriate MSW have been loaded with proper values, low level software takes over to carry out the motion. A pulse is issued to the selected motor causing it to advance 1.8 degrees. After a delay specified by the first entry in the delay table, a second pulse is issued, and the speed pointer is incremented to point to the second value in the delay table, and the acceleration counter is decremented. This process continues until the acceleration counter reaches zero. When this occurs, the acceleration phase is complete and the constant speed phase is entered. For this phase, the speed pointer is

Figure 5: Format of the motor status word. There is one motor status word allocated to each stepper motor; an interrupt driven process updates and times the operations of the motors. The information includes motor identification, status flags and parameters which specify the details of a cycle of operation consisting of acceleration, constant running speed and deceleration.

Motor Status Word (MSW)

Motor Number	Motor Flag
Speed Pointer	
Top Speed	
Total Steps	
Acceleration Counter	
Constant Speed Counter	
Deceleration Counter	

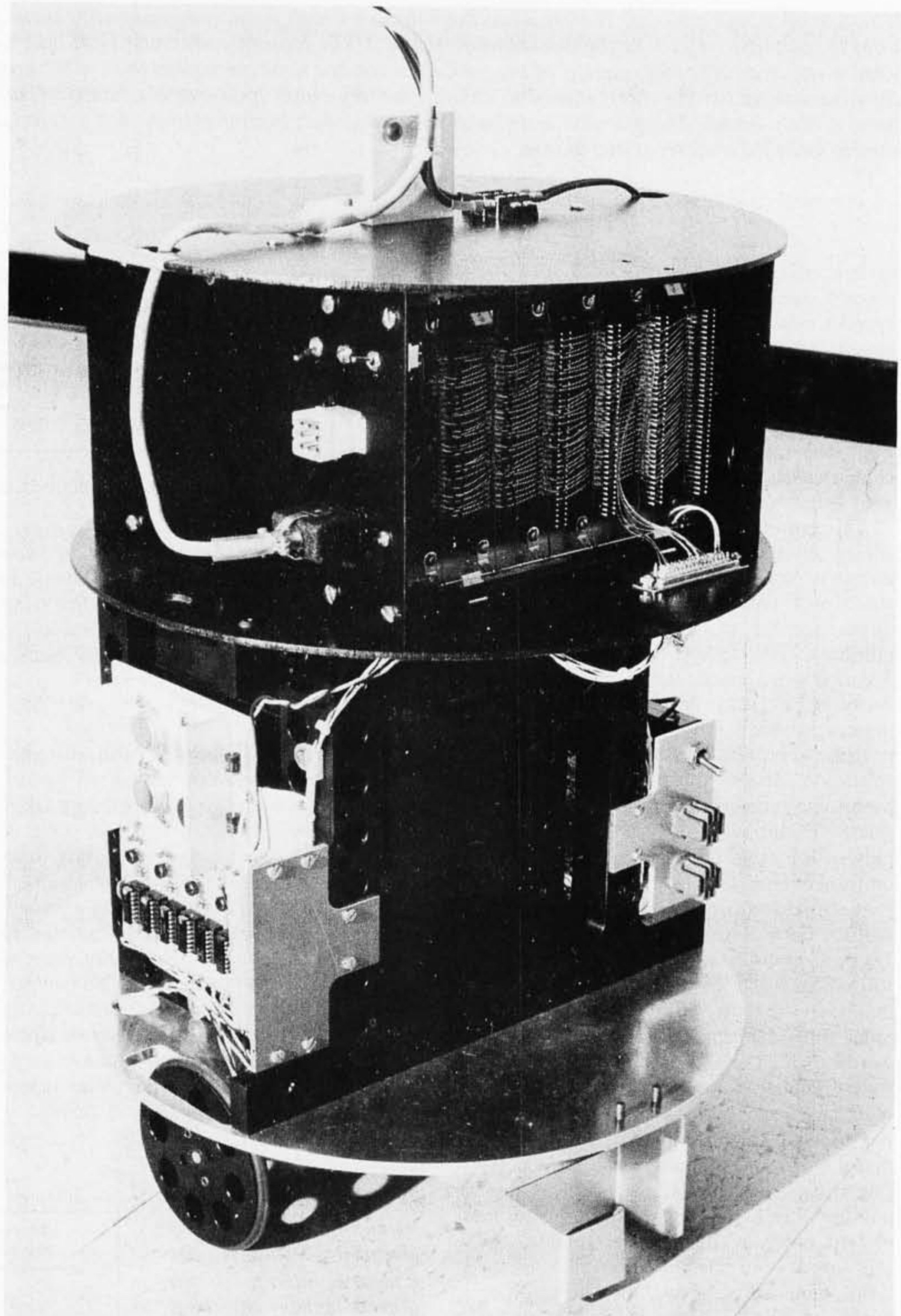


Photo 4: The robot Newt as it appears in the current state of construction. The hand wired computer backplane and stepping motor drive modules and sheet metal "hand" and power cables to an external source are all temporary.

held fixed, pointing to the time delay value corresponding to the top speed index, while the constant speed counter is decremented for each step. After the constant speed phase is completed, the speed pointer is decremented for each pulse, causing the motor to decelerate. The motor finally stops after a number of pulses have been issued equaling total steps. The motor is then placed in standby condition and this fact is stored in the motor flag byte. If unforeseen conditions arise which make it inadvisable or impossible to complete the motion as planned, the interrupt software stores the appropriate emergency stop code in the motor flag and the motor is halted. In all but extreme emergency cases requiring instant stopping of a motor running at high speed, the software can recover the total number of steps actually taken, saving it for navigational purposes.

To exercise direct control of the robot, a simple interpretive command system was developed. The currently implemented commands are listed in table 1. Each command consists of the 1 byte ASCII character F, B, L, R, W, V or Q, followed by a 2 byte argument. In the case of the F, B, L and R commands, the argument is simply the number of motor steps to be taken (less than or equal to $2^{15} - 1$). For the W command, the argument is the waiting time in units of 10ths of a second. For the V command, the argument specifies the new maximum speed of the robot in steps per second. For the Q command, the argument gives the 16 bit address of a sequence of commands stored in memory. Of course, many other commands, such as those appropriate for sensory turret and manipulator motion, will be added to the list of table 1. A program of robot activity consists of a simple sequence of these 3 byte commands, one after the other. For example, the following 15 byte sequence (arguments shown in hexadecimal notation):

```
F 0A10
L 000B
B 0010
W 0066
B 0002
```

would cause the robot to move forward 2576 steps (about 1.4 meters), turn left 11 steps (a few degrees), back up for 16 steps, stop and wait for 10.2 seconds, and then back up two steps. All accelerations and decelerations are taken care of automatically by the lower level software.

Commands can be given to the robot in several different ways (command modes) which are specified in table 2. For the D mode, the robot simply executes the commands directly as they are entered on the Teletype. After each command is given, it is necessary to wait until it is carried out before entering another command. In the R mode nothing is executed, but each command is recorded in sequence in the memory. The N mode is a combination of the D and R modes. It permits a sequence of commands to be recorded as they are being executed. The P mode allows playback of any previously recorded sequence. In this mode, the Teletype cable can be disconnected to allow the robot to roam freely. If the C mode is specified, the robot creeps along at a very slow pace under an F, B, L or R command for an indefinite number of steps. The motion is terminated by depressing any key on the Teletype, after which the number of steps taken is printed out. The T mode is a combination of the C and R modes. The E, L and O modes are "bookkeeping" in nature, and permit erasing, listing, and setting the origin of a command sequence, respectively. During the creation of a program of action for the robot (command sequence), any combination of modes can be used as desired. It is easy to think of many other useful modes which could be added to the list of table 2.

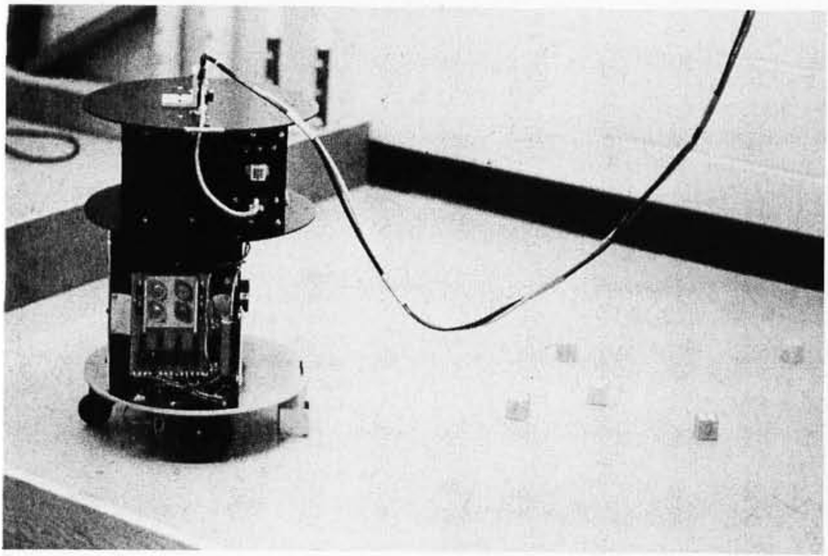
Using the commands listed in table 1, it has been possible to do a number of preliminary experiments with the robot which serve to test both the hardware and software, and to prove some of the fundamental ideas. For example, one can place wooden alphabet blocks on the floor at random, and then program the robot (using the N and T modes) to "pick them up" by trapping them in its temporary "hand." After putting the blocks and the robot back in their initial positions, the playback (P) mode can be used to repeat the motions automatically. One such test, rather in the form of a demonstration, is shown in photos 5a to 5g. In the first frame, photo 5a, the robot is moving towards the nearest block from its

Table 1: Currently implemented commands. The demonstrations of operation seen in photo 5 were created using an interpretive sequence including these seven basic commands. The command list is open ended in that many additions to the possible operations are expected as the software is developed further.

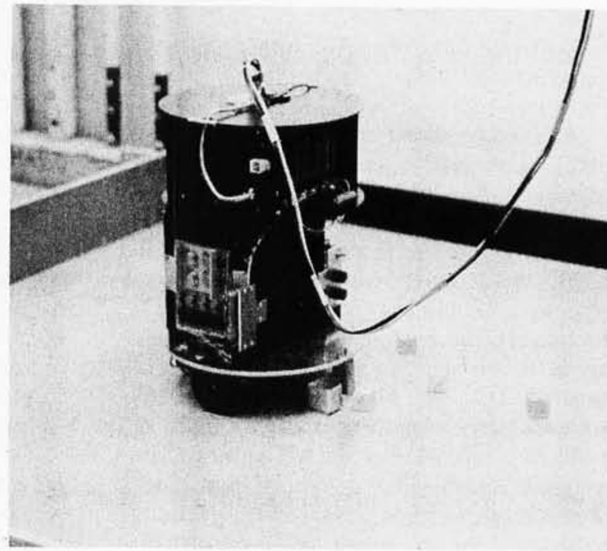
ASCII Command Code	Command Definition
F	Move robot forward
B	Move robot backward
L	Rotate robot left
R	Rotate robot right
W	Make robot wait
V	Set robot maximum speed
Q	Execute command sequence

Table 2: Command modes for the robot. The software presently implemented is used to try out various exercises of the robot's machinery. The software is structured into several command modes described by this list.

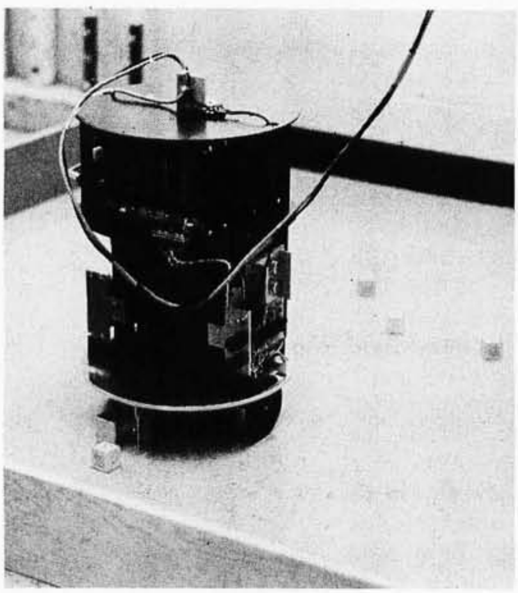
Command Mode	Type of Operation
D	Direct (simply execute the commands)
R	Remember (record commands)
N	Normal (combine D and R)
P	Playback (run through a command sequence)
C	Creep (directly execute motions at a turtle's pace)
T	Teach (combine C and R)
E	Erase
L	List
O	Set origin



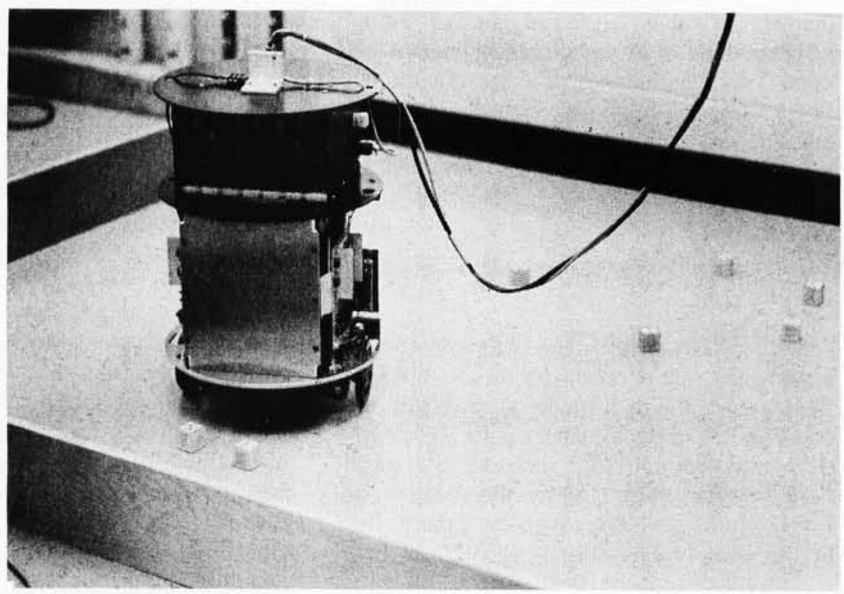
(a)



(b)



(c)



(d)

Photo 5: A program of action executed by the robot. In this sequence, prearranged blocks are pushed to appropriate locations to spell out N E W T, which is the robot's name.

(a) The robot's actions start with its movement from an initial position in the left corner of the playpen toward its first block.

(b) As it closes in on the block, it maneuvers into position to entrap the block. Note that in this prearranged course, the robot is executing a fixed sequence of interpretive commands, and is not yet using visual inputs to find blocks at arbitrary positions.

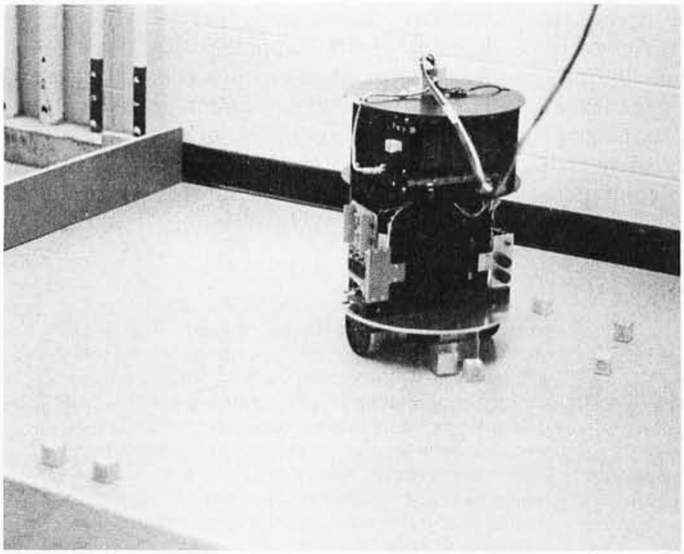
(c) After trapping the block and moving to the final position, Newt releases the block at its final position.

(d) Here, a second block has been fetched and released next to the first block, and Newt is turning around in order to head for the third block.

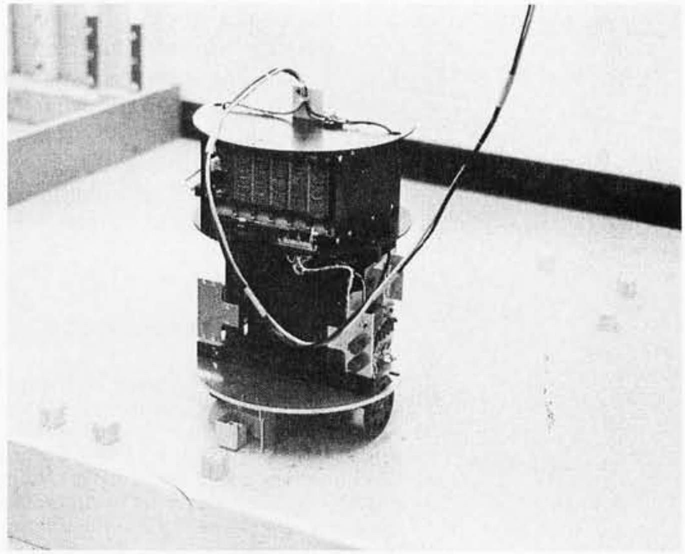
(e) Now Newt is ready to pick up the third block in the sequence.

(f) After placing the third block and going to the fourth, Newt has picked up this last block.

(g) Finally, the fourth block has been placed, completing the spelling of Newt's name, which is the end of the programmed sequence of approximately 100 interpretive motion commands.



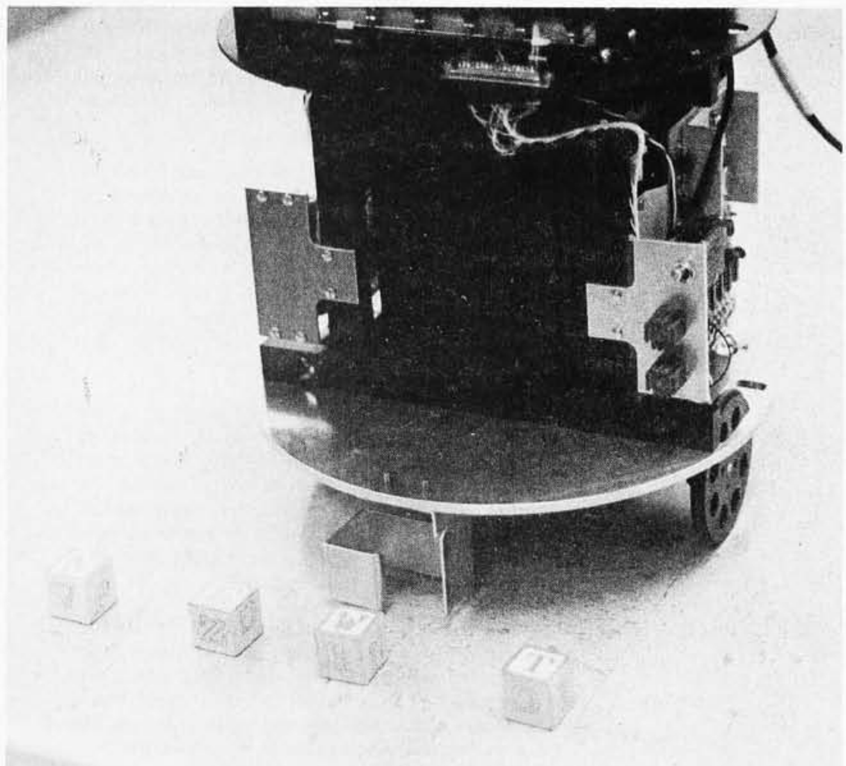
(e)



(f)

initial position in the left corner of the playpen. In 5b, it is maneuvering into position for "picking up" the block. In 5c, the block has been picked up, moved and released in a new position. In 5d, a second block has been fetched and placed next to the first block, and the robot is turning around and heading for a third block. In 5e, it is about ready to pick up the third block. In 5f, the third block has been placed, and the fourth block is in the "hand." In the final frame 5g, the fourth block has been placed, spelling out N E W T, which is, of course, the robot's name. Approximately 100 3 byte commands were required for this sequence. The program can be executed dozens of times without error, and the final positions of the blocks and the robot after the sequence is finished has a scatter of a few millimeters (±1/8 inch) in each direction. Of course, when the robot is complete with all its senses, it will seldom have to execute such a lengthy sequence in a completely open loop fashion. Perhaps it can eventually do tasks such as searching for and finding specific alphabet blocks in a random pile of blocks. This would require frequent closing of the feedback loops through the visual senses.

As the robot goes through its motions, such as depicted in photos 5a to 5g, it seems to possess an almost uncanny grace and precision. Small children, when watching it, are frightened at first, but this soon gives way to playful interest and warm curiosity. Even hardened computer experts are amazed to see a computer driving itself around on wheels!



(g)

Some Personal Remarks on Building Robots

It will not be possible for one person alone to write the software for the robot. This is far too large a job. It is hoped, however, that as the hardware nears completion, it will act as a focal point for many persons wanting to experiment with the robot by writing their own software. A fascinating project would be to create a general "Robot Control Language" which would free each programmer from the details of the hardware. What a rich experience it would be to work together, exchanging ideas in a highly interactive way. There are many tutorial possibilities for computer science classes and beginning students learning about computers. There are problems in psychology, procedural languages, human-to-machine communication, functioning of parts of the brain . . . (Working with robots is certainly one way to gain a much greater appreciation for the complex-

SOME TERMINOLOGY

Amp hour: A unit of energy for rating batteries. The battery of NEWT, with its 84 Amp hour capacity, can store enough charge to drive a steady 1 A load for 84 hours, or a 14 A load for six hours. The finite capacity of any practical battery means that any mobile robot must incorporate some programs for seeking electrical outlets and recharging batteries periodically.

Azimuth: As used here, an angle relative to a fixed direction in the horizontal plane.

Cognition: As Webster has it, this is "the act or process of knowing, including both awareness and judgement." *[Webster's New Collegiate Dictionary,* 1976 edition.*]* In the context of robots and artificial intelligence, this term refers to programmed models which approximate the behavior of natural cognition.

Degrees of freedom: The state of a robot mechanism (or any other system) can be described by specifying the current value of each variable parameter. Thus, if a robot arm has seven joints, the position of its "hand" might be determined by the angular setting of each joint. Each such independently variable parameter of a system is called a "degree of freedom," so the seven jointed arm would have seven degrees of freedom.

Heuristic: A heuristic computer program is one which starts out with an approximate method of solving of a problem within the context of some goal, and uses feedback from the effects of the solution to improve upon its own performance. Heuristic programming is one of the major contemporary artificial intelligence techniques, and is a key to developing a cognitive robot.

Manipulator systems: A generic term for any mechanical device which a robot uses to directly manipulate its environment. In the NEWT robot, this is currently (see photos) a simple sheet metal frame which can catch a block and slide it across the floor as the robot moves, with no active grasping; NEWT is intended to eventually have a much more flexible system of manipulation as described in the text. Most industrial robots currently in use consist of manipulators alone, without much in the way of sensory feedback or motive systems.

Motive systems: A generic term for the mechanisms used to convey the robot around its environment. In the NEWT robot, this refers to the two drive wheels, balancing caster and stepper motors which propel the robot.

Open loop, closed loop: A closed loop system is one which operates with feedback from errors. The feedback is intended to correct for the errors and thus approach the truth; an open loop system ignores error signals and operates on the sometimes naive assumption that no errors occur. The terms must be qualified by a reference to the time intervals involved in the system: NEWT, for example, is a closed loop system over long time periods, since it is intended to navigate using feedback from its sensors; however, due to the processing loads associated with sensors, NEWT operates open loop between navigation sightings in a manner analogous to the dead reckoning method of navigation used occasionally by airplane pilots or captains of ships.

Round off errors: In operations such as addition, multiplication or calculating transcendental functions, there is often some uncertainty in the least significant part of the result. In an extended calculation in which these operations are repeated over and over, appreciable round off errors can accumulate. In a digitally controlled vehicle guidance system such as that used for a robot, these numerical errors are a major source of uncertainty in the vehicle position, and are just as important as more obvious sources of error such as step quantization or slippage in the drive mechanism.

Senses: In a robotics context, senses are specialized peripherals which convert information about the environment into signals which can be analyzed by a computer or used directly by the electronics, as in a reflex. Sensory information may be obtained from devices as simple as a microswitch with a "feeler" arm, or as complicated as photoelectric imaging arrays with zoom lenses and pointing mechanisms.

Stepping motors: An ordinary electric motor is characterized by continuous motion when energized. A stepping motor uses a different design philosophy to achieve a motor which will move its shaft in small incremental angular steps on command, and will actively maintain its position in between each command. This type of motor is very well adapted to digital control of mechanical systems, and is used by NEWT for all mechanical motions in the robot.

Step quantization: The stepper motors have a finite angular resolution built into their design. This means that any mechanical motion derived from the motor will have a certain minimum step size, so that any attempt to position to a finer tolerance must be approximated.

Trajectory: The path of a moving object is its trajectory. In the case of the mobile robot, a trajecectory is planned before motion takes place, given a desired goal position and a world model which covers its course and objects which may be in the way.

World model: A world model is the result of cognition as implemented in robots. Formally, it is an information structure built up in the memory of the robot, based on both initialization and heuristic interaction with the environment.

ity and capability of the brain.) In addition, experiments can continue on sensor development and interfaces from sensor to computer. There are a great many practical spin-offs from this kind of work.

Many people believe that as more and more advances are made in microelectronics, the prospects of mass producing robots will become attractive, and the prices of these hypothetical machines will plummet. (Let us hope we will have learned something from *R U R*.) If this occurs, many applications will open up. Besides such things as planetary surface exploration, such as already demonstrated by the Viking robots, one can envision undersea robots working on oil pipelines and well heads, coal mining robots, fire fighting robots, agricultural robots, robots on assembly lines producing customized articles, robot-like prosthetic devices, and many other types of robots for specialized and general service, doing jobs which are too difficult, too dangerous, or which are otherwise undesirable for humans.

I have been asked many times the questions, "Why are you building a robot?", and "What will it do when it is finished?" The answer to the second of these questions is easy: I simply don't know what the robot will be able to do. This is the whole point of building the robot. Given a modest amount of hardware and a greater amount of software, thoroughly integrated to form a system, the idea is to find out just what such a system is capable of doing. The whole is likely to be far greater than the sum of the parts. The system is pushed as far as the available time, money and energy will allow in order to learn what can be done and what cannot be done; in other words, to explore the frontier of robot research, and to know and understand the problems involved. The necessary knowledge can neither be obtained by theoretical studies, nor by simulations using large computers.

In regard to the first of the questions, I have been fascinated with robots since the mid-1950s and have constructed several robot devices prior to the one described in this article. The construction of such machines presents many interesting challenges. A functioning robot is a most curious blend of electronics, mechanics, computer design, computer programming and artificial intelligence. All these fields come together in the design and construction of a robot, and each must be explored in depth. Added to this are the challenges and excitement of locating obscure components in surplus store parts bins around the country, planning, building, and then replanning, rebuilding, and, of course, experimenting and learning. To me, all these are fascinating aspects of endeavors which are perhaps best left to amateurs. I say this, because I strongly believe that the amateur computer enthusiast has a golden opportunity to participate in advances in the field of robotics. In fact, the amateur has several advantages over the professional. The research can be as abstract as the amateur wishes it to be and can be conducted without regard to immediate payoff potential in the marketplace. There is no need to spend time writing elaborate proposals, no need to continually justify the direction of the work, and no need to get hard results every few months to write up and stick into quarterly reports. History has shown that precisely this atmosphere of freedom which surrounds the amateur is the atmosphere in which brilliant innovations and discoveries are sometimes made.■

About NEWT's Name and Family Tree

The origin of NEWT's name is buried in an often quoted verse from *Macbeth* by William Shakespeare . . .

In the cauldron boil and bake;
Eye of newt and toe of frog,
Wool of bat and tongue of dog,
Adder's fork and blind-worm's sting,
Lizard's leg and howlet's wing,
For a charm of powerful trouble,
Like a hell-broth boil and bubble.

Newt I, the present robot's predecessor, was a light-seeking robot consisting of a large eye on a stalk rising above a motor driven platform.

Acknowledgment: The author wishes to express gratitude to Dennis Toms for his enthusiastic help and interest in this project.

Photo 1: The complete setup of the IBM Selectric Keyboard Printer, typing under the control of a KIM-1 microcomputer with a 4 K memory expansion. The Selectric interface described in this article is housed in the equipment case in the center of this photo.

Interfacing the IBM

Dan Fylstra
Hamilton Hall C-23
Harvard Business School
Boston MA 02163

Photography by Carole Brock

One of the most desirable forms of computer output is high quality typewritten text suitable for preparing letters, reports and other documentation. A word processing system which speeds up the process of writing and revising text would be a very useful and feasible application for a small microprocessor based system, provided that a suitable hard copy output device can be found at a reasonable price.

An ideal output medium for such a word processing system would be an IBM Selectric office typewriter. Selectrics are moderately expensive when compared to ordinary typewriters ($630 to $830 depending on the options chosen), but they are ubiquitous in the office environment, produce very high quality typed output, and can be used to print in many different type styles simply by changing the ball shaped typing element. Special typeballs are available for printing mathematical symbols and for the APL character set (see "What is APL?", by Mark Arnold, November 1976 BYTE, page 20).

Unfortunately, the job of converting a Selectric office typewriter is made somewhat more difficult by the fact that (contrary to popular belief) the Selectric mechanism is entirely mechanical and not electronic in nature. The only use of electric power in an ordinary Selectric is for the motor which turns the drive shaft and various gears and cams. It is necessary to use solenoids to push levers and "bails" in the base of the mechanism to achieve printing under computer control. Similarly, contact switches must be installed in order to use the keyboard for computer input.

There is another alternative, however. A variety of computer terminals and other devices based on the Selectric mechanism are becoming available on the surplus market, often at a fraction of their original prices. These machines have their own built-in solenoids or other means for mechanical control, and present some sort of electrical or electronic interface to the outside world. The simplest, most commonly available, and of-

Selectric Keyboard Printer

(Teaching KIM to Type)

ten the cheapest of these are the Selectric Input/Output Keyboard Printers, Models 73, 731, 735 and others. They were manufactured by IBM, typically for use as IO devices in other companies' computer systems. As these systems have become obsolete, the Selectric Keyboard Printers have found their way into surplus channels.

As a business school student and experienced user of computers, I have always wanted to build a word processing system around my own home computer. Hence I seized a chance to acquire a Model 73 Keyboard Printer for $450 from the Computer Warehouse Store in Boston. (These units were sold out in a few weeks; I have heard of prices ranging from $250 to $1500 through other channels, but as interest in the units increases, their typical prices are bound to rise.) Armed with a couple of old IBM manuals provided by the Computer Warehouse Store, I set out to accomplish what I expected would be a straightforward interfacing process.

This article is a report of my experience, and a detailed description of the interface which I built. Briefly, the interfacing process, while simple in principle, was not at all straightforward in practice. But it was successful, even for such a mechanically inept person and relative novice in electronics as me. For about $50 in parts (including such extravagances as a pretty cabinet and a $20 IBM connector to plug into the Selectric's peculiar 50 pin receptacle), and lots of labor, I produced the unit shown in photo 1. It's only an interface to the Selectric printer, since I'm content to use my existing ASCII keyboard for input. It has its limitations, but it works.

This, of course, is hardly the last word on Selectric Keyboard Printer conversion. As a BYTE reader, I would be delighted to see information on more comprehensive interface designs, as well as actual experiences with several of the units currently on the market. Since most of them are sold on an "as is" basis, these machines can bring

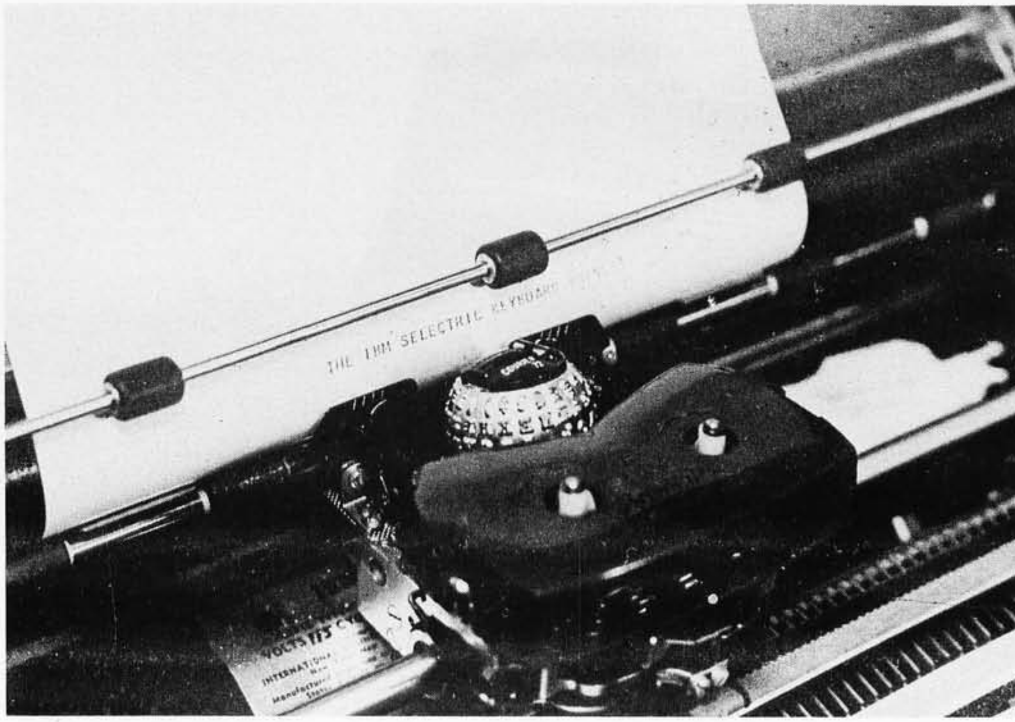


Photo 2: A closeup of the IBM Selectric ball mechanism on its moving carriage within the printer. The Courier 72 ball shown here is one of many balls available with the "Correspondence" coding arrangement.

you a lot of grief (read on). So it's wise to report on problems, and how you overcame them.

The Selectric Mechanism

To appreciate what the interface must do (and what can go wrong), it is first necessary to understand basically how the Selectric mechanism works. The typing element is a golfball sized hollow sphere embossed with up to 88 characters, arranged in four horizontal rings of 22 characters each. Photo 2 illustrates the ball in its rest position in the mechanism. All the lower case or unshifted characters appear on the "western hemisphere," the side normally closest to the paper. The upper case characters are in corresponding positions on the back side or "eastern hemisphere." Pressing the shift key causes the whole typeball to rotate 180°, thereby allowing the upper case characters to be printed. Hence, the actual typing operation can select any of 44 characters, four half rings of 11 characters each, with five to the left and five to the right of the center or "home" position on each ring. A particular character is selected by causing the typeball to tilt up or down and rotate right or left; then the ball jumps forward to strike the ribbon and paper. These movements account for the peculiar "dancing" motion seen when the Selectric is typing continuously. The typeball is mounted on a carriage which moves across the page, as opposed to traditional pre-IBM typewriters where the paper carriage moves and the typing mechanism remains stationary.

The actual tilting and rotation of the typeball is accomplished by an incredibly complicated system of latches, pulleys and levers which are driven by six moving "bails," or rods in the base of the machine. Although we need not understand the detailed mechanical linkages, we should appreciate the roles played by these six

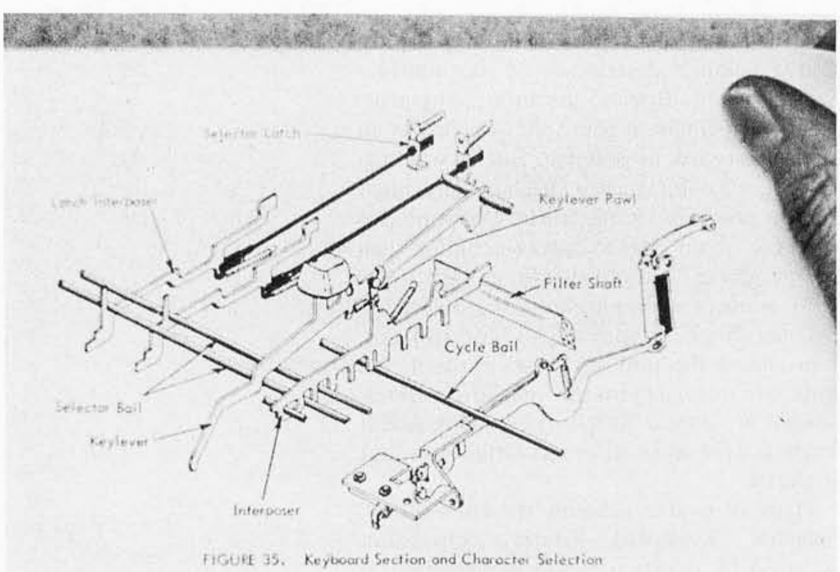
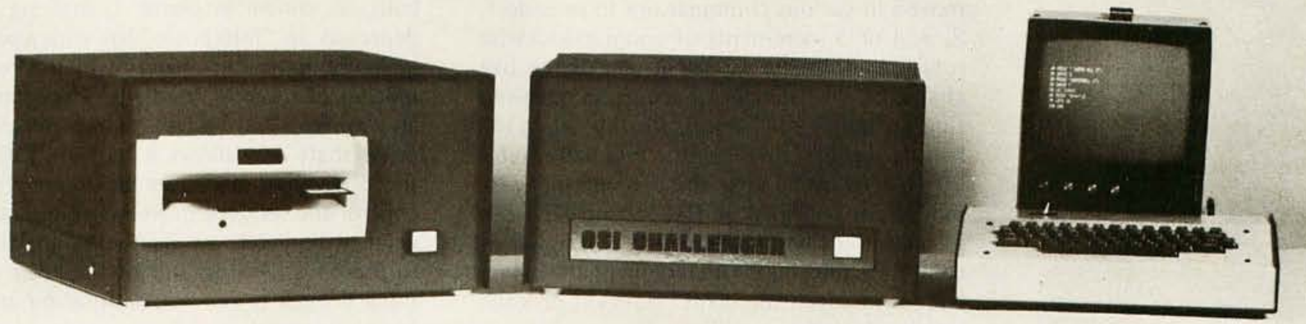


Photo 3: This diagram, from the IBM manual number 241-5159-3, shows how the various bails of the mechanism are connected in a typical case. The IBM manuals for the typewriter output unit are valuable reference materials and can be obtained by calling your local IBM office.

INTRODUCING THE COMPLETE CHALLENGER SYSTEM.



OSI's DREAM MACHINE
is a totally integrated computer system.

Imagine a system complete with terminal, CPU, memory, floppy disk, software, and all the little necessities to make it work together immediately. Now imagine this complete system available not only fully assembled, but priced much lower than anyone else's kit. What you are dreaming of is OSI's "new" Challenger System!

In the configuration shown above, the Challenger includes everything an end user needs for a complete small computer system. All you add is 110 VAC power and a desk to put it on.

This fully-assembled system includes:

HARDWARE:

OSI Challenger 65 with 16K RAM, serial interface, system monitor PROM, and floppy disk bootstrap PROM.

OSI Challenger single drive floppy disk formatted for 250K bytes storage per diskette surface.

Stand-alone terminal and Sanyo monitor for 16 lines of 64 characters at 2400 baud (other terminal options are available).

And all interconnecting cables!

SOFTWARE:

2 diskettes containing over 100,000 bytes of software including OSI's powerful Disk Operating System with variable length sectors. **6502 DISK BASED RESIDENT ASSEMBLER/EDITOR!** A totally interactive Assembler/Editor which assembles up to 600 lines a minute and is completely compatible with MOS Technology's Cross Assembler format. This program also contains a powerful disk-based line editor with commands for general text editing. **OSI'S EXTENDED MONITOR:** A powerful machine language debugging and utilities package including a Disassembler which is format compatible with the Assembler! **OSI 6502 8K BASIC FOR DISK BY MICROSOFT:** This powerful BASIC has all the features of Altair™ 8K BASIC for the 8080 plus higher speed and disk storage. And it comes complete with a BASIC program library.

DOCUMENTATION AND SUPPORT:

We include over 600 pages of hardware, software, programming, and operation manuals. The Challenger is based on the well-proven OSI 400 system. The over 2,000 OSI 400s and Challengers now in use assure continuing hardware and software support for this system for years to come!

EXPANDABILITY:

The Challenger System can now be expanded to 192K of RAM and 16K of I/O and ROM. There are over 13 accessory boards including A/D, D/A, parallel and serial I/O, cassette interfaces, a dual drive floppy, a video graphics display, several RAM and PROM boards, and multiple-processor configurations.

APPLICATIONS:

The Challenger system is complete, fully assembled and configured so that the Disk Operating System can be booted in immediately on system power-up. Even a relatively inexperienced operator can have a complex BASIC program on-line just seconds after the system is turned on. The ease of use, high reliability, and large library of standard BASIC applications programs make the OSI Challenger System the first practical and affordable small computer system for small business, educational institutions, labs, and the personal computerist.

PRICES:

Challenger System, complete as stated above with terminal and monitor

$2599.00

As above without terminal. Specify RS-232 or 20ma loop and baud rate

$2099.00

IMPORTANT NOTE:

One of the most important features of the Challenger System is that it is not really "new". OSI has been delivering the basic circuitry of the Challenger since November 1975 and the floppy disk since June 1976. The only thing new is the total integration of the components as a complete, simple to use, fully-assembled, small computer system.

For more free information and the address of the OSI Computer Dealer or representative in your area, write to: OSI; Dept. S; Hiram, Ohio 44234 or enclose $1.00 for the full OSI catalog which contains kits from $134 and fully assembled computers from $439.

OSI

Ohio Scientific Instruments

11679 Hayden Street, Dept. S, Hiram, Ohio 44234

Circle 40 on inquiry card.

moving bails. Two of the bails, referred to as T1 and T2, are moved or not moved in one of the four possible combinations to provide the proper degree of tilt necessary to select one of the four rings. Three more bails, called R1, R2 and R2A, are moved or not moved in various combinations to provide 1, 2, 3, 4 or 5 increments of counterclockwise rotation, normally to select one of the five characters to the right of center on the given ring (as seen from above). Finally, when the bail named R5 is moved, the typeball rotates 90° clockwise so that the counterclockwise movement provided by R1, R2 and R2A can select one of the five characters to the left of center on the ring. (When none of the rotate bails is involved the center position on each ring is selected.)

To print a particular character, then, we need to know its position on the typeball (which can vary from ball to ball), as well as what combination of bail movements – T1, T2, R1, R2, R2A and R5 – will take us to that position. Figure 1 presents the "coordinates" of each character in terms of the six bail movements for the two most common character arrangements, the ones used on the "BCD" and "Correspondence" encoded typeballs.

The Keyboard and Print Magnets

In an ordinary Selectric typewriter, the keys are mechanically linked to the various bails, as shown in photo 3. Striking a key depresses an "interposer" bar with a particular combination of fingers which arrest the motion of some of the bails. The interposer also moves a "cycle bail" which releases the drive shaft and allows it to turn 180°. On the drive shaft are a number of cams which control the series of movements necessary to print a character, as selected by the tilt and rotate bails. At the end of the cycle everything is back to normal, waiting for another key to be struck.

In a Selectric Keyboard Printer, the tilt and rotate bails are also mechanically linked to six electromagnets. The magnets pull down armatures which otherwise would arrest the motion of the bails. To print a character, some combination of the six magnets must be energized, the particular tilt and rotate "code" for that character as found in figure 1. In addition, something

TYPEHEAD LAYOUT

T – #s = Tilt latches active
R – #s = Rotate latches active

	Upper Case											Lower Case											Tilt	
	–5	–4	–3	–2	–1	0	+1	+2	+3	+4	+5	–5	–4	–3	–2	–1	0	+1	+2	+3	+4	+5	T2	T2
BCD TYPEBALL CODES	■	■	■	■	@	>	■	■	■	:	√	1	3	5	7	8	Ø	2	4	6	9	'	0	0
	■	■	■	‡	%	\	■	■	■	⌢	#	/	T	V	X	Y	ŧ	S	U	W	Z	,	1	0
	■	■	■	|	✳	;	■	■	■	]	△	J	L	N	P	Q	–	K	M	O	R	S	0	1
	■	■	■	?	¤	>	■	■	■	[	‡	A	C	E	G	H	&	B	D	F	I	.	1	1
ROTATES R5	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0		
R2A	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	1	1		
R2	0	0	1	1	1	0	0	1	1	1	1	0	0	1	1	1	0	0	1	1	1	1		
R1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	0	1	0	1	0	1		
CORRESPONDENCE TYPEBALL CODES	[	'	&	*	$	Z	@	%	¢	)	(	]	3	7	8	4	z	2	5	6	0	9	0	0
	X	U	D	C	L	T	N	E	K	H	B	x	u	d	c	l	t	n	e	k	h	b	1	0
	M	V	R	A	O	°	.	''	I	S	W	m	v	r	a	o	j	.	'	i	s	w	0	1
	G	F	:	,	?	J	+	P	Q	Y	_	g	f	;	,	/	j	=	p	q	y	—	1	1
						HOME											HOME							

COLOR KEY

= Home Position of Ball
= Tilt Rotate Command Bits

Figure 1: The graphic fonts of the BCD and Correspondence typewriter codes. The location of each character on the Selectric ball is described by a unique combination of case, tilt and rotate commands. The upper case versus lower case choice is made by a mechanical latch set up before printing, so the chart is broken down into two main sections for each code. The home position of the typeball is flagged in each case by a color shading. The binary command information for each matrix position is given by the rows in the center labeled "ROTATE" and the columns at the right labeled "TILT." Thus, to form the Correspondence code for the letter S, the tilt command bits are 01, and the rotate command bits are 0110 for a tilt rotate command code of 010110 to be used in the format described in figure 7.

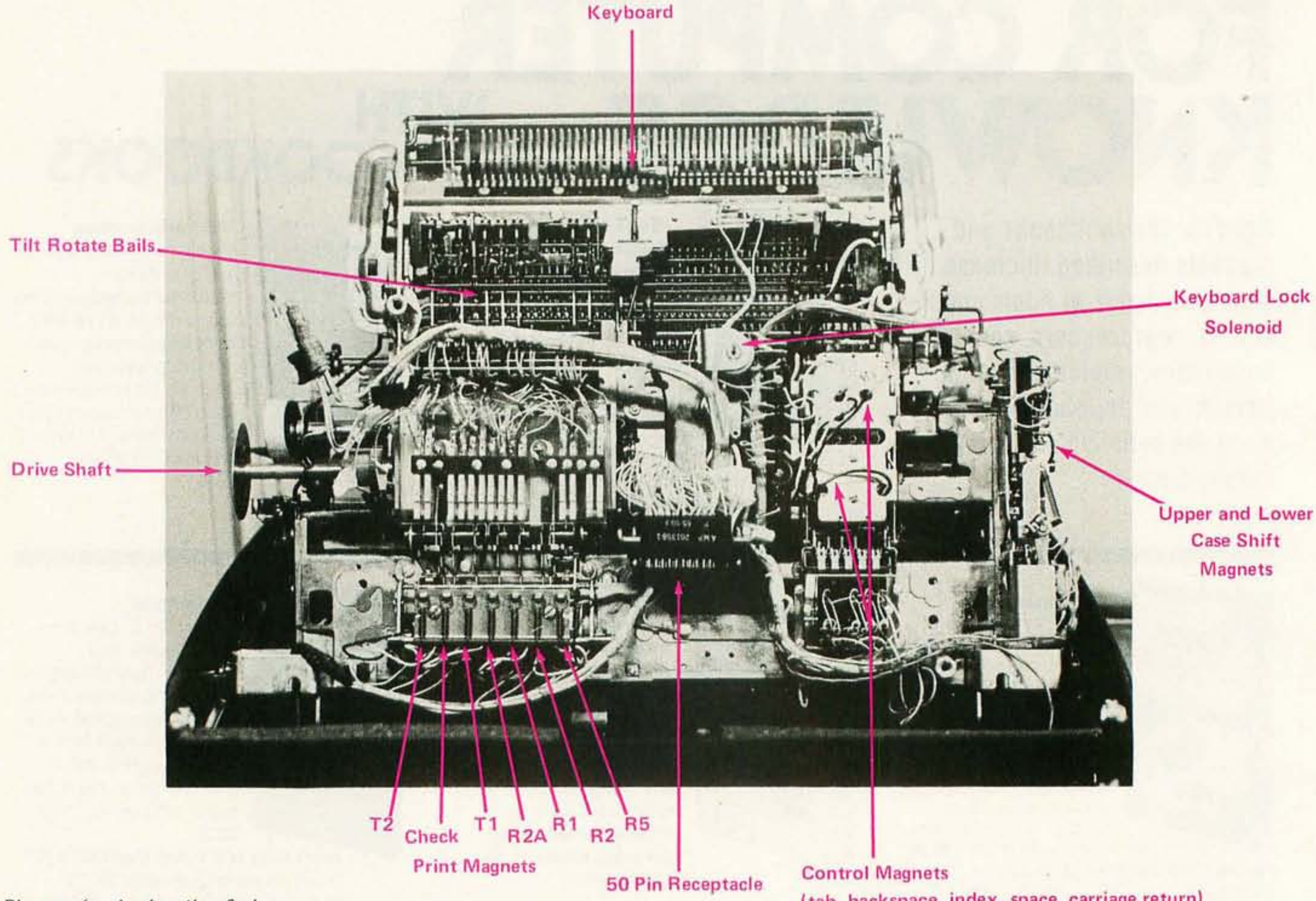


Photo 4: A detail of the underside of the Selectric Keyboard Printer with housings removed. The overlay shows several of the key points such as the location of various magnets, the switch contacts and interconnection receptacle.

must actuate the "cycle bail" to start the printing process. Hence a trip mechanism is provided which moves the cycle bail whenever any of the armatures is pulled down. However, there is one character on each hemisphere which should be printed when none of the magnets is energized, for the code 000000. Hence the trip mechanism is connected to a seventh magnet, called "check," which provides an odd parity function for the other six magnets. It is energized whenever necessary to ensure that the total number of magnets energized is odd. Thus the check magnet is energized on the code 000000, and this serves to actuate the cycle bail. (I didn't realize this when building my interface, so I can't print those two characters yet. Don't make the same mistake!)

Besides the print magnets, there are a number of other magnets and armatures inside the Keyboard Printer which control special functions such as space, backspace, tab, carriage return, index (ie: advance paper without returning), ribbon shift, and upper and lower case shift. Many of these magnets can be seen in photo 4, which exposes the underside of the machine and outlines the positions of many components. The upper and lower case shift magnets are latching, and hence they lock the machine into the new case until the opposite magnet is energized. Note that the operator cannot shift the machine back into lower case when the upper case magnet is latched! By Murphy's Law this is bound to happen whenever you are testing the interface, but it can be remedied by fooling around with the shift cam at the end of the drive shaft.

No electric power is provided for any of these magnets inside the Keyboard Printer, but the coil connections are brought out to the 50 pin receptacle at the back of the machine. The magnets are rated for 43 to 53 VDC at 125 to 300 mA, applied for at least 10 ms in order to pull down the armatures and cause the desired action.

Switch Contacts

The other major addition to the basic

Continued on page 133

Joseph J Carr
Bioelectronics Laboratory
The George Washington Univ Medical Ctr
901 23rd St NW
Washington DC 20037

Interfacing With an Analog World—Part 2

Last month we discussed transducers and amplifiers. These are necessary portions of a signal processing system which result in scaled voltages of, for example, 0 to 10 V corresponding to the original physical parameter being measured. But how can we convert these voltages into numbers inside a computer for computation, and use numbers from computations to control external voltages? In this article we'll see how some of the more common conversions are accomplished. We'll start with digital to analog conversion, even though this may seem at first glance to be backwards. The reason for starting with the output process is that digital to analog conversion is simplest, and that many analog to digital input conversion techniques require a digital to analog conversion as part of the process.

There are several techniques for making a digital to analog converter (often abbreviated DAC). In most cases these converters use some form of binary weighted current or voltage summation that is controlled by the digital word presented to its inputs. A typical example is the classical R-2R technique which is outlined in figure 1. The active element A1 is an operational amplifier of unity gain inverting follower configuration. Although an amplifier of the 741 general purpose family will suffice in many experimental situations, it is often better to select one of the more costly premium grade integrated circuit operational amplifiers. We have shown the digital inputs of the circuit in figure 1 as mechanical switches, a configuration which is most useful in a tutorial situation when teaching the concept of the digital to analog conversion. In practical digital to analog conversion applications the switches are electronic, and are controlled by some form of n bit binary data source such as a counter or computer output port.

A precision reference voltage source is required as well, and for many commercial and industrial system designs this will be a precise 10.000 . . . V. The accuracy of the converter is largely dependent upon the precision of the reference voltage. Although any precise voltage regulator circuit may be used for the reference, I have found in laboratory instrumentation experience that commercial products such as the REF-01CJ or REF-01HJ (for severe applications) work quite well. In a homebrew experimenter's situation, where relaxation of tolerances is quite normal, a simple zener diode reference circuit will often prove quite adequate.

Now let's consider the circuit in figure 1 in more detail. What happens when various combinations are presented to the digital input? Suppose that all the input bits are in the low state, which means that they are connected to ground by the electronic or mechanical switches shown. The value of the output voltage is given by the product $I_1 R$. When all bits are grounded through the switches, the input current to the amplifier is zero, as can be deduced by tracing, noting that there are no nonzero inputs to the amplifier's feedback node. (In practical circuits, though, there may be some output voltage under these circumstances due to offsets in the operational amplifier itself. These undesirable offsets may be nulled out through an offset adjustment potentiometer arrangement, R_{NULL} in this circuit. See some of the tutorial design books on operational amplifiers for further elaboration of this detail.)

If the most significant bit of the word is

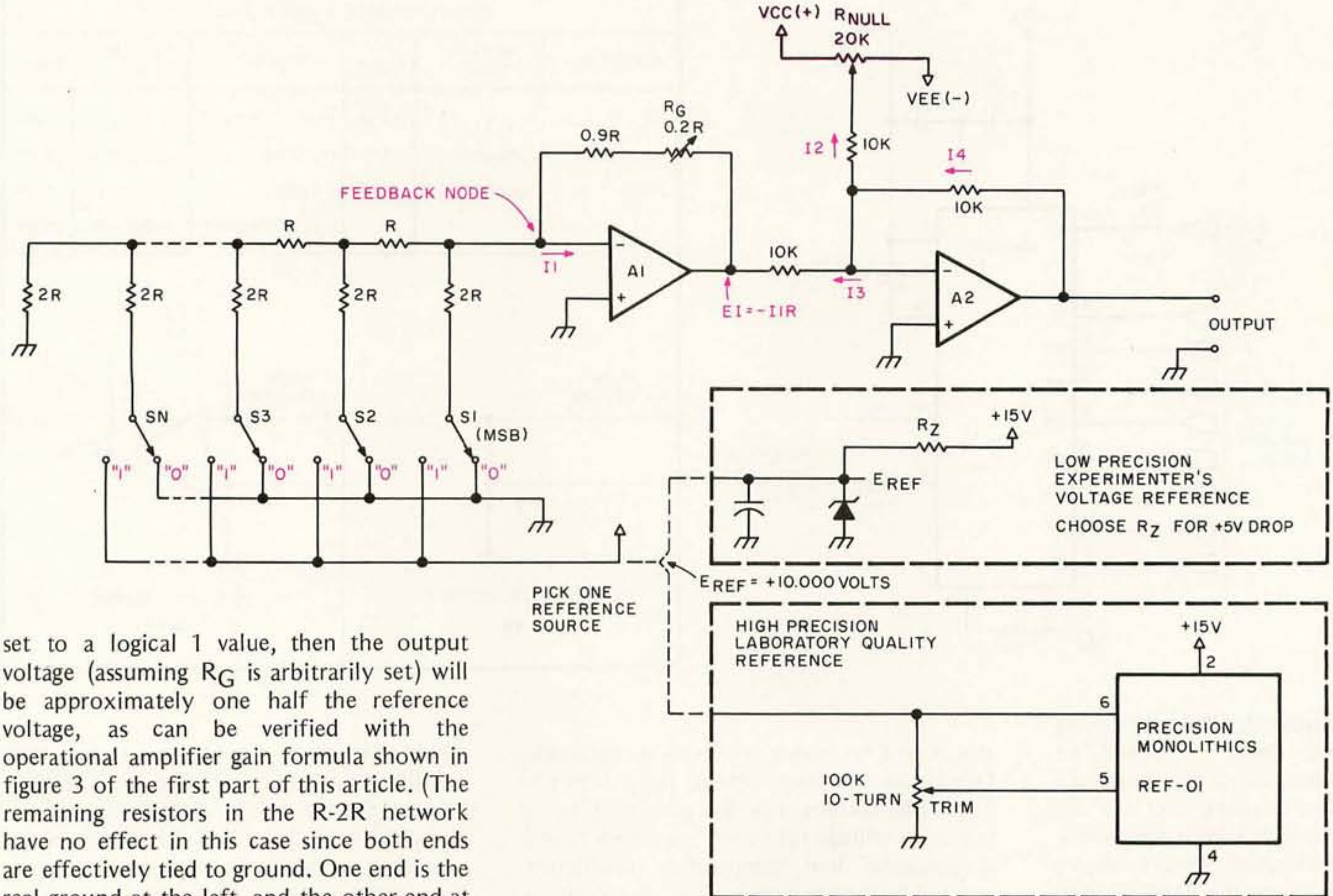


Figure 1: A classical R-2R network digital to analog converter implemented as a circuit diagram with discrete parts and operational amplifiers. The essentials of any digital to analog converter are present: a reference voltage (two alternatives shown), a switched network that creates a binary weighted current controlled by the switches, and buffering and conversion amplifiers to create a voltage output which can drive other circuits.

set to a logical 1 value, then the output voltage (assuming R_G is arbitrarily set) will be approximately one half the reference voltage, as can be verified with the operational amplifier gain formula shown in figure 3 of the first part of this article. (The remaining resistors in the R-2R network have no effect in this case since both ends are effectively tied to ground. One end is the real ground at the left, and the other end at the feedback node of the amplifier is its "virtual ground" for the signal.) The analysis of the next most significant bit and the remaining bits of the digital word is a bit more complicated, but the result is what might be expected. The bit controlled by S2 in figure 1 will contribute one fourth of the reference voltage to the output of A1; the bit controlled by S3 will contribute one eighth of the reference voltage. And for switch, or bit, n (where n starts at 1) the contribution will be $E_{REF}/(2^n)$.

Let us assume that we have an 8 bit digital to analog converter of the type shown in figure 1. The word at the input terminals is 11001011 and the reference voltage is precisely +10.000 VDC. What is the output voltage? The following calculation, which is easily generalized, shows how the value is derived:

$$E_o = 10 \times \left(\frac{1}{2^1} + \frac{1}{2^2} + \frac{0}{2^3} + \frac{0}{2^4} + \frac{1}{2^5} + \frac{0}{2^6} + \frac{1}{2^7} + \frac{1}{2^8} \right)$$

Output (E_o); E_{REF} (10); Fraction Based on a Digital Word (the bracketed sum)

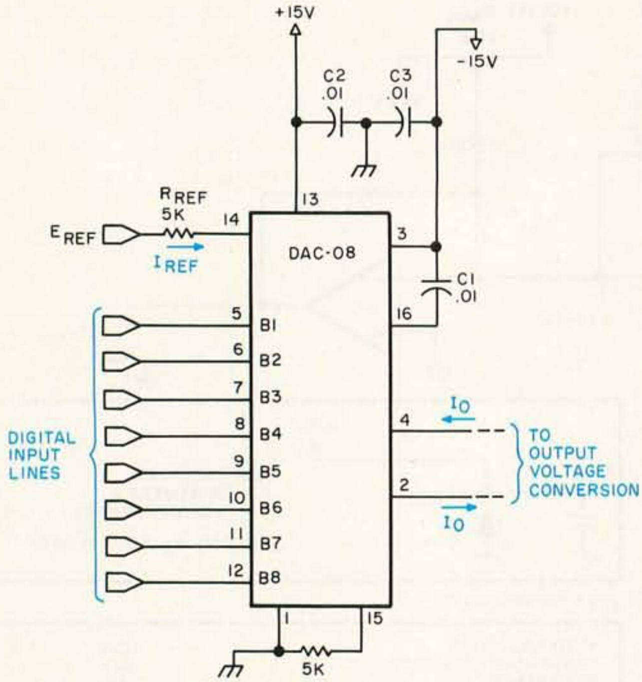
$$= (10/2) + (10/4) + (10/32) + (10/128) + (10/256) =$$

$$= 5 + 2.5 + 0.3125 + 0.078125 + 0.0390625$$

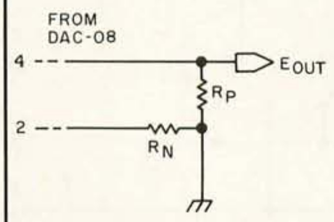
$$\simeq 7.93 \text{ V}$$

But use of amplifiers and resistors as shown in figure 1 is hardly optimal in an age of integrated circuits. A number of manufacturers offer convenient low cost 8 bit integrated circuit digital to analog converters that contain almost all of the electronics, except possibly the E_{REF} supply and the operational amplifier used for output voltage conversion and level shifting. I have used those by DATEL, Analog Devices and Precision Monolithics with good results in laboratory instrumentation. Experimenters and designers will also find the parts in the Motorola MC1408 family, as well as several similar parts made by Signetics, to be quite useful. For my examples in this article I have selected the Precision Monolithics DAC-08. I found this product easy to obtain in low quantities (ie: one) through local distributors.

Figure 2 shows the basic circuit for using the DAC-08, along with two voltage conversion schemes for its current outputs of pins 2 and 4. The integrated circuit itself contains the electronic switches, the resistance ladder, a reference amplifier and the current output buffer that drives pins 2 and 4. Two types of input are required to make

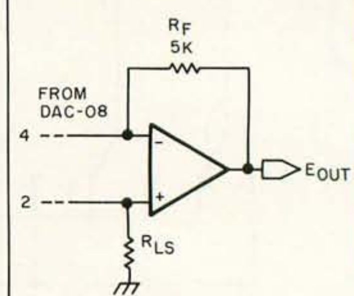


OUTPUT VOLTAGE CONVERSIONS					
CONDITION	BINARY CODE	E_{OUT}	CONDITION	BINARY CODE	E_{OUT}
FULL SCALE	111111111	-9.96	(+)FULL SCALE	11111111	+9.96
HALF SCALE	100000000	-5.00	(+)ZERO	10000000	+0.04
ZERO	000000000	0.00	(-)ZERO	01111111	-0.04
			(-)FULL SCALE	00000000	-9.96



NEGATIVE OUTPUT

4B



BIPOLAR OUTPUT

4C

Figure 2: While the circuit of figure 1 could be constructed, it is usually more convenient to replace it with a monolithic integrated circuit device such as the DAC-08 part by Precision Monolithics, which is used for this illustration. The integrated circuit contains all the components of the R-2R network of switches and buffering amplifiers. It still requires an output conversion circuit (two variations shown) and a reference voltage. Because the R-2R network is an integrated circuit, this form guarantees the highest possible accuracy with none of the extra bother of hand wiring a circuit as in figure 1.

this digital to analog converter do its work. One is the reference current, I_{REF} thru pin 14. This current can be generated by a precision voltage reference (see figure 1) and a precision low temperature coefficient resistor in the ideal cases. (For low precision applications, ordinary resistors will work just fine.) For the configuration shown in figure 2 and a reference potential of 10 V, this resistor should be precisely 5000 ohms, a number which is derived from the documentation of the DAC-08.

The second major type of input to the converter is the 8 bit digital word that is applied through pins 5 to 12. In the notation of figure 2 and throughout this article, the bits are numbered from 1 (most significant) to 8 (least significant). Bit 1 of the DAC-08 package is wired to pin 5, with bits 2 thru 8 wired to pins 6 thru 12. The logic levels at the inputs are the usual TTL levels, with a low voltage (approximately 0 volts) signifying a logical 0, and a high voltage signifying a logical 1.

One of the output conversion circuits shown in figure 2 is a simple unipolar conversion which uses two resistors and no operational amplifier. With this conversion, the formulas discussed in figure 1 apply except for the fact that the voltage is negative with respect to ground. When the input word is binary 00000000, then the output of the converter system is 0.00 V. Half scale (−5 V output) is given by an input word of binary 10000000, and full scale output occurs when the input word is binary 11111111. The output under full scale conditions will not be 10 V, but approximately 9.96 V. This slightly unexpected condition is due to the mathematics of the switching network. Evaluating the formula for the conversion given earlier, with a reference of 10 V and a binary 11111111 digital value, we find:

$$E_{out} = 10 \times (255/256) = 9.96 \text{ V}$$

Also shown in figure 2 is an output circuit which uses an operational amplifier as a level shifter and voltage conversion device. Wired with the components shown, this amplifier connected to the DAC-08 gives a gain of 2 and shifts the generation of output levels to a symmetric bipolar range of −9.96 to +9.96 V. Note that it is impossible to get an exactly zero voltage in this case, since the 256 possible states are split symmetrically about zero. If the level shifting reference resistor R_{LS} were adjusted slightly off the 5 K value, the voltage range of the conversion could be pulled slightly (ie: 0.04 V) positive or negative so that a true zero would be possible for one of the binary states. There are other possibilities for the output conversion circuits and as in any design situation, a little bit of imagination always comes in handy. *[Readers looking for more examples of typical applications should consult the applications notes of the various manufacturers. Of particular use is the excellent specification sheet and application notes on the MC1408 DAC, published by Motorola . . .* **CH**]

Analog to Digital Conversions

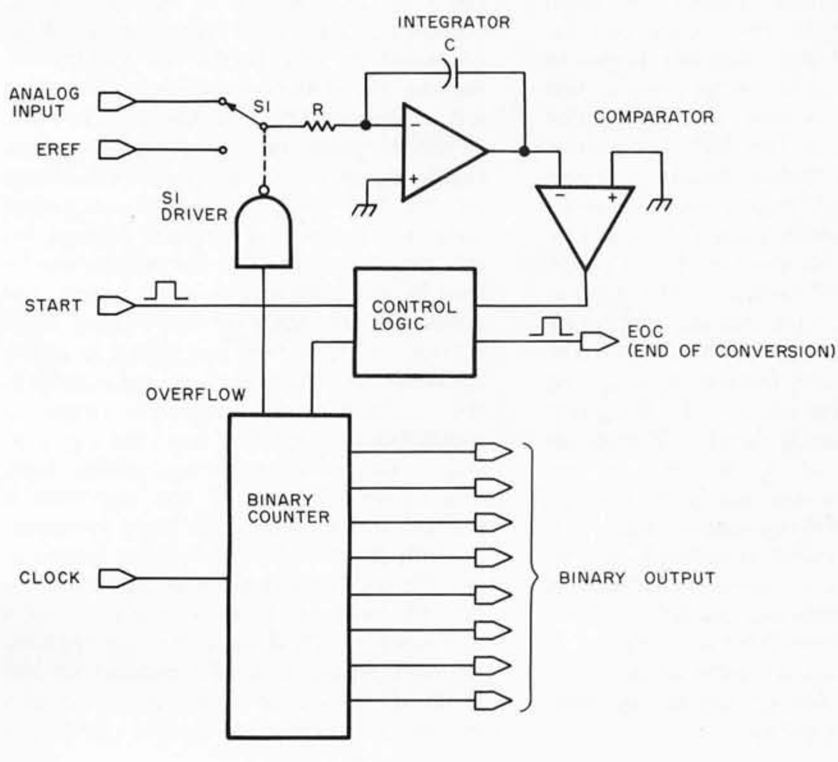
With the concept of a digital to analog conversion covered, it is now possible to consider the opposite case: conversion of measured voltages from the sensor preamplifiers into numbers which can be processed and used by a computer. Of the many techniques which are available for performing analog to digital conversions, we will only consider the details of integration, counter (or ramp), and successive approximation methods here. These are the simplest and most universal methods.

One of the basic parameters to be considered when talking about any analog to digital conversions is speed. This is not a major consideration in the output problem already discussed, since digital to analog current output conversions essentially take place at the switching speeds of digital logic, and are then limited only by the final operational voltage output amplifier's response. In the input case, however, some form of approximation cycle which converges upon the digital value is required; as a result, the conversion can be somewhat slower.

Integration Methods

At the slow end of the analog input conversion spectrum is the integration method. This is the type of conversion which is typically used in digital panel meters and similar instrumentation. These can be useful in cases where you might mount the digital panel meter or multimeter in a system, both as a readout mechanism and as a measurement conversion device. Many such instruments offer parallel digital outputs on their rear panels, along with control and strobe lines. The appeal of this approach often is affected by two characteristics: relatively slow conversion speeds and binary coded decimal (BCD) encoding directly taken from the displays. The relatively slow conversion rates become a problem when looking at signals other than "slowly varying DC levels" of very low frequency sources. The coding characteristics may in fact be optimal for many computational schemes in a computer program, but it can be a nuisance if one attempts to use such a meter in a binary oriented hardware system. The typical "dual slope" integrator used in these digital panel meter circuits is illustrated in figure 3.

The dual slope conversion circuit consists of five basic sections: an integrator, a comparator, a control logic section, a binary counter, and a reference current or voltage source. The integrator consists of an operational amplifier connected with a capacitor in a negative feedback loop. This capacitor is charged by the operational amplifier output voltage. The input to the integrator is taken from either the analog input or the reference source. The comparator is made with an operational amplifier that has an open feedback loop.



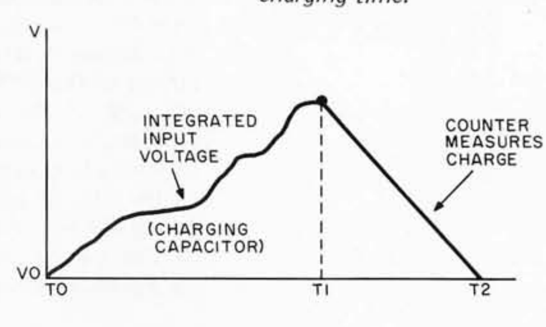


Figure 3: Analog to digital conversion by the dual slope integration method is often performed by slow devices such as digital panel meters. This method works through an analog integrator and a counter. The integrator has switchable inputs. It first integrates the incoming signal for a specified time interval. Then it counts the time necessary to linearly discharge the charge just accumulated with a known slope. The result is a count which is proportional to the voltage which drove the integrator during the charging time.

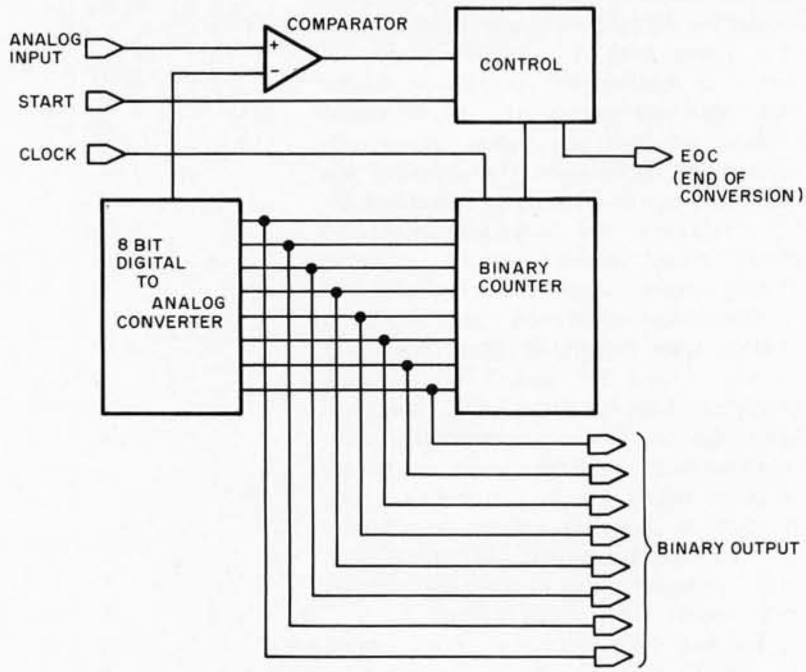


Figure 4: The ramp or counter method of analog to digital conversion is straightforward: A counter is initially zeroed and then allowed to count upwards until its binary code through a digital to analog conversion creates a voltage greater than or equal to the input voltage.

This makes its gain very high. If the two input voltages are not equal, then the operational amplifier output is high or saturated. In this case the comparator is ground referenced and uses just one active input.

When a START command is received the control circuit resets the counter to 00000000, resets the integrator to 0 V output (discharges C), and sets switch S1 to the analog input. The analog voltage creates an input current to the integrator which causes the integrator output to begin charging capacitor C. This means that the output voltage of the integrator begins to rise. As soon as this voltage rises a few millivolts above ground the comparator output snaps high. The high comparator output causes the control circuit to trigger the counter, which begins counting clock pulses. The counter is allowed to overflow and this outputs an overflow bit. This bit changes the state of switch S1. The graph in figure 3 shows the integrator charging during the interval between START and the overflow of the binary counter (t_0 to t_1). At time t_1 the switch changes the integrator input from the analog signal to a precision reference source. Also, at time t_1, the counter has overflowed and again it has an output of 00000000 (maximum count + 1 is the same as the initial condition). It will, however, continue to increment so long as we have a high comparator output.

The charge accumulated on capacitor C during the first time interval is proportional to the average value of the analog input voltage between t_0 and t_1.

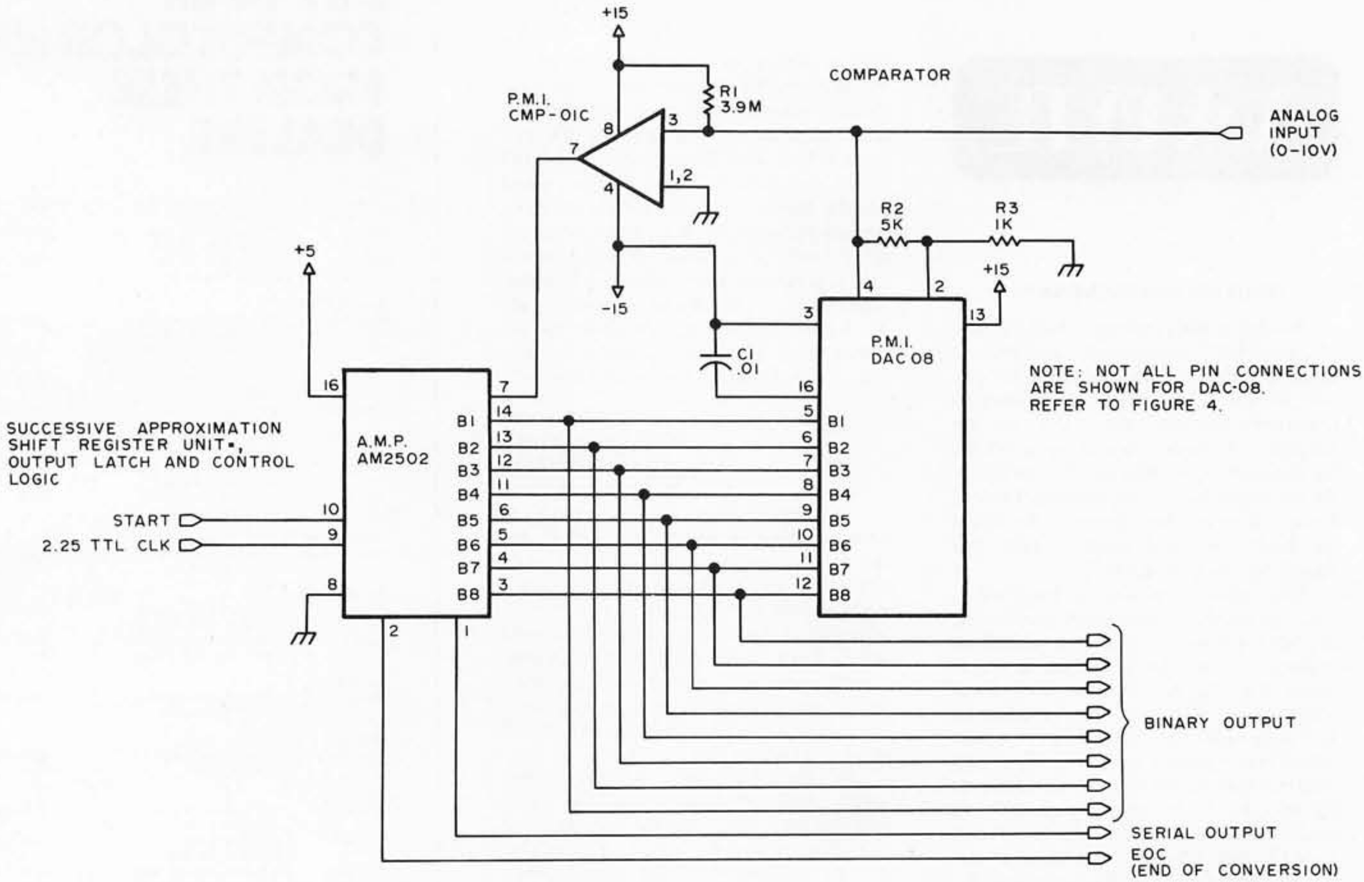
Capacitor C is discharged during the next time interval (t_1 to t_2). When C is fully discharged the comparator will see a ground condition on its input and again it will change state to make its output low. This causes the control circuit to stop the counter but *does not reset the counter.* The binary word at the counter output at the instant it is stopped is proportional to the average value of the analog waveform over the interval t_0 to t_1. An end of conversion (EOC) signal is generated to let the microprocessor or other system know that the output data is stable, valid and ready for use. The speed limitations are based on the requirement for an accurate and stable analog integrator, and the need to average the input signal over a long cycle.

Counter (Ramp) Analog to Digital Conversion

A counter or ramp type analog to digital conversion circuit is shown in figure 4. Here we find a voltage comparator, a digital to analog converter with voltage output, a binary counter, and necessary digital control logic. Here is how the conversion works:

When the START command is issued by a control system (for example, a computer output port) the control logic resets the binary count to 00000000 and enables the clock input to the counter, which begins counting upwards at the clock rate. Since the counter outputs set the voltage level out of the digital to analog converter, the DAC generated voltage to the comparator will begin to rise. As long as the analog input voltage is greater than the reference voltage generated by the digital to analog conversion, the state of the comparator output will enable counting via the control logic. However, when the rising reference level finally equals or exceeds the input voltage for the first time, the comparator output state will switch and stop the counter. At this time, the output of the counter can be read by the computer or other system, and represents the value of the analog input voltage. If the counter and digital to analog converter are both eight bits, the number in the counter will represent from 0 (00000000) to 255/256 times the digital to analog converter's reference voltage level. The conversion time of this algorithm is proportional to the voltage being measured.

Both the dual slope integration technique and the counter technique discussed thus far take far too long for many applications. On the order of 2^n clock cycles are typically required where n is the number of bits involved. Conversion time becomes critical in an application when the frequency



response must be high and a faithful reproduction is required. (For reference, consult textbooks in electrical engineering concerning the "sampling theorem" and Nyquist's criterion that says we must have a sampling rate of at least twice the highest frequency that is to be recognized, if a faithful reconstruction of the signal is to be achieved.)

Successive Approximation

In programming and software design, we often find use of a "binary search" stratagem instead of a flat out sequential search when trying to speed up the process of finding an item in a table. This same approach is just as applicable in hardware, where the "successive approximation" technique of analog to digital conversion provides a much higher speed of conversion than the sequential counting methods discussed so far. The successive approximation technique typically requires only n+1 clock cycles to make an n bit conversion, and its hardware is no more complex than that of the dual slope or counter methods.

The successive approximation converter, shown in figure 5, consists of a comparator, control logic, a shift register with output latches for this form of conversion, and a voltage output digital to analog converter. When a START command is issued to the converter circuit, it loads a binary 1 value into the most significant bit of the shift register, which in turn sets the most significant bit of the output latch. This sets the output of the digital to analog converter to half scale. In true binary search strategy, if the input voltage is less than the reference output provided by this half scale setting, the most significant bit is cleared from the latch on the next clock pulse; otherwise, the most significant bit is left unchanged at the next clock pulse. Then the internal shift register of the successive approximation register unit is shifted so that its single high level bit is opposite the next most significant bit. Again, the output register is modified, this time so that bit 2 is set to 1 for the trial measurement. This bit has a value of one fourth of the total voltage range, which is added to the half range or zero value still latched from the first measurement. At the next clock pulse, if this new trial value to the comparator is greater than the input value, the 1 bit is latched in the successive approximation register; otherwise a 0 bit is inserted at the current position. This process continues with successively less significant bits until the shift register overflows indicating that the last bit has been tested. Some forms of this conversion have control logic to detect an equality condition and

Figure 5: The successive approximation method, here illustrated with a practical circuit, uses a binary search strategy. The most significant bit is tested first, then the next most significant bit, and so on down the line until all n bits have been tested. If at each stage the contribution of the selected bit causes the trial approximation output from the converter to exceed the input value, the bit will be stored as a zero. After all n bits have been tested, the result is an n bit binary representation of the voltage of the input.

Continued on page 62

Ask BYTE

BRITISH COMMENTARY

As one of the relatively small number of personal computing addicts in England, I'm very impressed with the rate at which the field is growing in popularity on the other side of the Atlantic. I've been subscribing to BYTE for the past six months and I promise it's always made fascinating reading. I wish I knew where in this country one could lay hands on any issues before last August's. (Hint, anybody?)

The advertisements are fascinating, too. Anybody considering purchasing any sort of system is almost forced to import it from the USA. My query is about TV displays that most of these systems use. Naturally, they're designed for American TV standards, which differ from British ones. We have 625 interlaced lines per frame, repeating at 25 Hz. Is it, in general, possible to modify an American TV display device to work with a British TV? Or is all the timing generation usually performed inside a special chip? Might I have more success if I tried to modify a TV instead? I know rather more about TTL logic than about televisions, but I'm game to learn if necessary. Otherwise, if the answer to all these questions is no, then the temptation to come and live in the States is going to be almost unbearable . . . .

I look forward to every issue of your magazine; I'm only sorry that I discovered it so late. Best wishes for the successful future that you deserve.

Guy A Burkill
Corpus Christi College
Cambridge CB2 1RH ENGLAND

Are there any other British readers who have experienced the problems of interfacing American video generation logic to European television designs? Some firsthand knowledge of the subtleties to be found would be the best way to answer Guy's query on that subject.

With regard to back issues of BYTE, there are none left. However, much of the editorial content is now available in a book titled Best of BYTE *edited by David Ahl, publisher of* Creative Computing. *This book contains reprints of numerous articles from the first 16 issues of BYTE, September 1975 thru December 1976.* ■

SACRED BUSES

I am very curious about something. What is sacred about the Altair bus as opposed to others? Would it not be possible to install a peripheral designed for the Altair scheme to, for example, a Digital Group or Southwest Technical Products bus? I am confused on the issue since it seems to me that, functionally, lines must correspond between the systems pretty well. Perhaps the answer lies in the fact that the Altair bus was based on the 8080 processor, which has some unique IO methods. Is that the problem? Is it really a question of 8080 versus other processor compatibility? I understand why a manufacturer would want to make plug compatible cards to the Altair bus, but why couldn't a simple conversion be available for any product?

J C Chirigos
1601 Kentucky NE
Albuquerque NM 87110

The main issue is one of plug-in compatibility. Whether you call it the Altair bus, or, as used by non-MITS suppliers, the S-100 bus, the key to the wide availability of the peripherals is nominal compatibility at the hardware level. Even here, there are occasional clashes between various manufacturers about the definitions of pins not originally assigned meanings in the Altair definition of the bus.

At the detail level, a 100 pin bus surely works, and in principal one could talk to any other similar computer at the memory and bus interface level simply by simulating all the signals which would have been present on the bus in its Altair implementation. This is a quite workable procedure, as demonstrated by many products.

True, the IO structure of the 8080 is unique, and there are some 8080 specific features of the Altair bus as a result of this and other characteristics of the 8080. But for the most part, the particular selection of the lines present on the 100 pin interface of the Altair bus is just a reflection of discretionary choices on the part of the designer(s) of the first Altair within the framework of the general design of a microprocessor with 16 address bits and eight data bits. ■

SOME TERMINOLOGY

I have a problem.

First, let me give you some idea of my background. I have spent 15 years in the computer industry as an applications programmer. I am fluent in BASIC, FORTRAN, PL/I and APL. Now that I have left the industry, I am getting interested in recreational computing.

I have never been involved in logic design or in the details of hardware, nor do I feel that I want to get involved with it now. I would not mind building a kit provided that the instructions were of the "put tab A into slot B" type. However, when I read all the ads and literature furnished by the various manufacturers, I have the following problem: What are you all talking about?

I thought I knew what "read only memory" was. Now I come across "pro-

Continued on page 67


STAR TREK

Continued from page 59

stop the conversion ahead of overflow, but the worst case time is the n+1 clock pulses for n bits mentioned earlier. Figure 6 shows the sequence of voltages presented to the comparator by a 6 bit successive approximation algorithm, as compared to a 6 bit counter conversion algorithm to show the time savings of this method.

In the circuit of figure 5, the comparator is shown as the Precision Monolithics CMP-01C device, although faster conversions may be possible if a higher speed Advanced Micro Devices AM686 comparator is used instead. The Advanced Micro Devices AM2502 integrated circuit, which was designed for this successive approximations conversion application, contains everything needed for the logic described verbally above, except the digital to analog converter and the comparator. Other companies go even further with integration of the input conversion. Precision Monolithics, for example, makes an AD-02 circuit which contains all the complete 8 bit analog to digital conversions. It is relatively expensive, but its cost can often be justified by its utility and ease of use. It has a respectable conversion speed and has several input options that can accomodate analog voltage ranges of 0 to 5 V, 0 to 10 V, −2.5 to +2.5 V, −5 to +5 V, and −10 to +10 V.

Software Approaches . . .

As noted earlier, the software of a microprocessor can often implement the algorithms of digital to analog conversion. This is especially so with the successive approximation algorithm, since its inherent speed makes up for some of the slow facts of life concerning programmed execution. To rig a software approach to the problem, we need a digital to analog converter attached to an output port, an input comparator which drives one input line, and the software of successive approximation (or other methods for that matter). In this case, the successive approximation shift register is variable in a program, the output latch is an output latch connected to the DAC device, and decisions are made based on the single bit input. A previous article in BYTE *[see "Microprocessor Based Analog/Digital Conversion," by Roger Frank, page 70 of May 1976 BYTE]* discussed both the ramp (counter) and successive approximation methods described here, but showed how to implement them in software.

Whether the approach taken is that of pure hardware or software aided designs, adding analog input conversions to a personal system can expand its capabilities to cover many interesting real world control and measurement problems.■

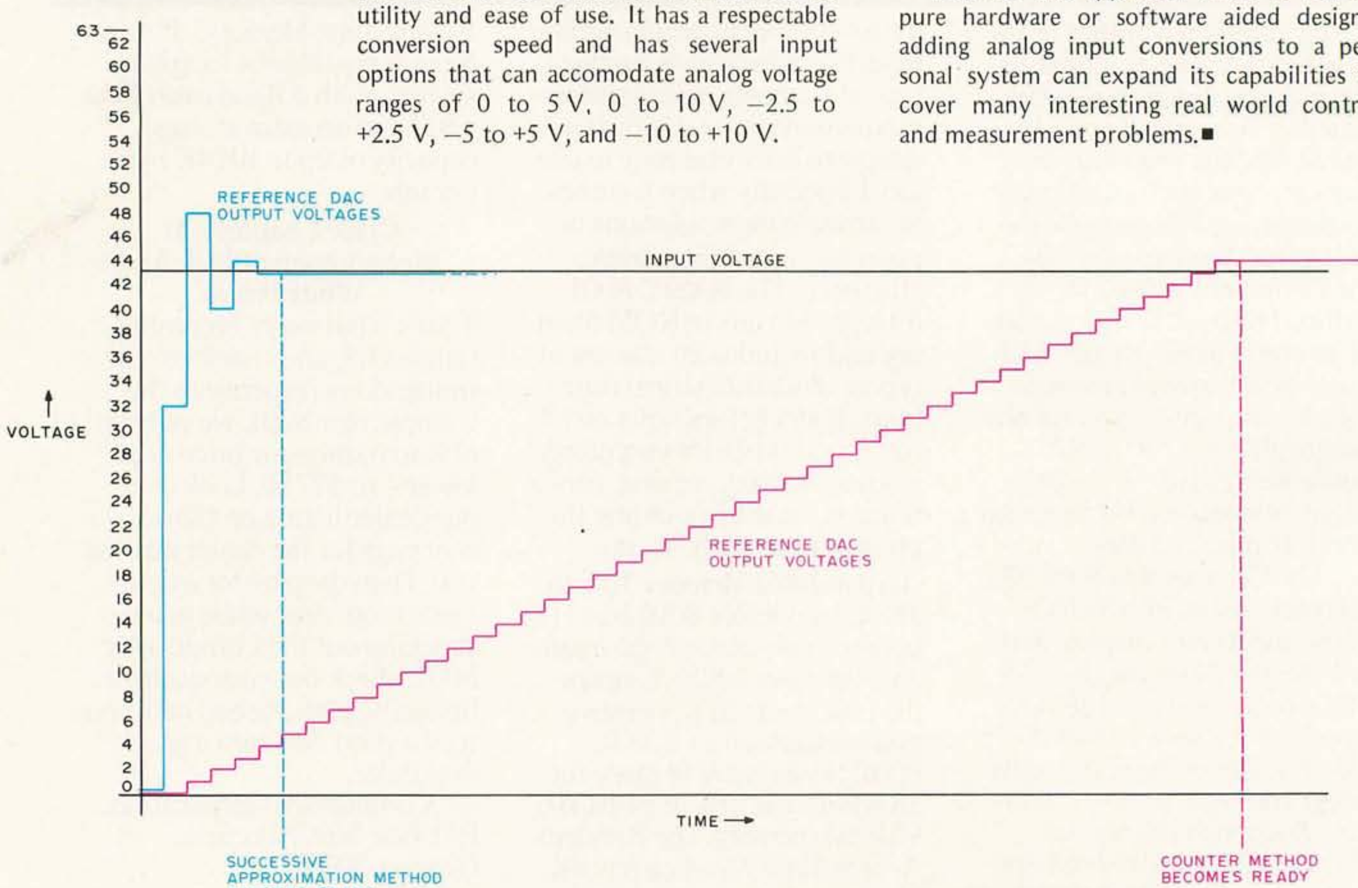


Figure 6: Comparison of the successive approximation method with the ramp method of conversion for a 6 bit value. (Six is chosen for purposes of this illustration.) The successive approximation method takes a mere six cycles of testing to arrive at the best value, where the ramp method has to count up to the number which matches the input and causes the conversion to terminate. The ramp takes 44 cycles here, versus six for the successive approximation method.

Personal Computing
National Computer Conference
June 13-16, 1977

rayson/sketch pad

Continued from page 60

grammable read only memory" (or worse yet, erasable programmable . . .). If it's read only, how can it be programmable? What is static memory? Is there any other kind of memory other than "random access memory?"

Is it possible for you to help me out of my dilemma? Something between "binary numbers are made up of zeros and ones" and "when the static EPROM is connected to the DMA using a 3P+S IO module."

Al Weiss
POB 942
Alleghany CA 95910

Short of the tutorial article which may be inspired by your questions, here are a few notes on terminology and concepts in computer design and implementation.

Random Access Memory: *For the purposes of discussing the present state of technology, a random access memory means some sort of reference to semiconductor memory parts (excluding serial access devices such as shift registers). A magnetic core memory is a form of random access memory, but is not used in modern small computer technology due to various manufacturing and economic considerations. A read only memory (ROM) is a random access memory part, as is an ordinary programmable memory. The read only memory is distinguished from the fully programmable memory by the fact that it is nonvolatile (the information is retained when power is removed) and moderately difficult or impossible to alter once it has been set up with a program.*

A commonly used acronym found in technical jargon and advertising is RAM, referring to random volatile programmable semiconductor memories. Since both read only memories and volatile programmable memories are random access devices, this term is misleading and ambiguous. In BYTE's editorial content, we do not use the term RAM intentionally, because of this ambiguity, and refer instead to "programmable" and "read only" memory. Programmable *emphasized the volatile, user program oriented nature of parts which are often called RAMs in conventional engineering journals, and* read only *characterizes the nonvolatile, permanent nature of the other type of random access memory part. (Of course, "programmable" is still not an optimal choice, since even read only memories are always programmable in the sense of "program it only once.")*

Static Versus Dynamic: *In brief, there are two types of volatile programmable memory parts, characterized by the internal design of the basic memory cell of the circuit. A static memory typically has a cell with sufficient active transistor elements to create a true flip flop memory register. A dynamic memory typically has a smaller memory element size which is achieved by replacing the memory flip flop with a capacitor which stores an electric charge. As a result of this smaller unit bit cell size, at the limits of technology the largest dynamic parts have historically had about four times as many bits as the largest static memory parts. At the present time, the largest static memories readily available are 4 K bits per chip, whereas the largest dynamic memories are 16 K bits per chip.*

There is a subtlety of design with dynamic memories, however. This is the fact that since the storage elements are capacitive in nature, sensing and support electronics on the chip tend to drain the charge with time, losing any information stored in the cell. The dynamic memory chips must thus be refreshed periodically, an operation that is commonly performed by cycling through a reference to the low order bits of the address inputs to the chip. Static memory chips have no such refresh requirement, and are often easier to use in prototype, small or homebrew circuitry; the manufacturing economies tend to make dynamic memories the most attractive in larger systems products.■

Continued from page 9

number of existing computer science textbooks. But I'd really like to somehow buy the design in a completely documented form so that like the plan of the hypothetical rolltop desk, I could implement it literally with custom modifications. Tutorial and "how to" plans books for specialized fields such as those mentioned above are widely available already, and at prices well within the range of an individual's budget. They are marketed in large quantities because large numbers of individuals use the information; outlets range from mail order book services to retail stores. Drawing out of this parallel between individualized computing and individualized "anything else," it should be obvious what the solution of the software dilemma is: Publish detailed plans and tutorial information for software, on a scale commensurate with the size of the market. Publishing ideas is an activity which has a long and distinguished history, and yields both personal and financial rewards to those who engage in the practice, as well as real benefits to those who purchase the products. Let's turn now to the application of this concept to the software designs of the computing world.

The Ideal Model of a Published Software Product

When we talk about publishing software at the present state of technology, we are talking about a product which is akin to the detailed design of the rolltop desk mentioned earlier. It is a product which serves as the starting point for the home software craftsman, not a recipe which will fit without thought into every conceivable system. This will change a bit as the systems in the marketplace become more refined, but the nature of the computer as an intellectual amplifier tends to require a certain level of technical familiarity on the part of its user. (This is the element which distinguishes the general purpose computer from the applications oriented dedicated computer such as a 4 function calculator or oven controller.)

In order to make a software package which is optimally configured for the customer's standard or customized use, there is a certain minimal level of documentation which is required. This level of documentation is not necessarily needed by all users all of the time, but is in many respects akin to the reference books for integrated circuits: When a question needs to be answered, it is good to have the information needed to zero in on the answer. Here is what I consider to be adequate documentation:

- Users' manual textual materials concerning the "standard" uses and limitations of the software. Here is where we find such information as standard IO patch points, relocation tables, etc.
- Complete object code, preferably machine readable along with machine readable relocation information.
- Complete source listing of the package including source language and generated object code for each statement.
- Program logic manuals and tutorials on the design of the product are an excellent option.

The idea is to include enough information to allow the user to do routine field alterations, including relocation. The idea of a published software product is to compile all this information together in a comprehensive book form, to be sold at prices characteristic of books, as opposed to the past history of software prices for applications and system software packages. The technology of printing covers all the portions of the "complete" package except machine readable code, at least in the minds of most people. However, as we have demonstrated with experiments published in BYTE, printing technology also covers machine readable representations as well.

Varieties of Machine Readable Representations

User convenience demands that a software product be made available in some form of machine readable representation. While it is certainly possible to take an object listing in printed form and type it into a processor by hand, this is a long, tedious and error prone process. To complete the functional definition of "adequate documentation" given above, we need a form of machine readable object code at minimum, along with machine readable relocation information. Fortunately there exist several technologies which can be employed for this purpose, which I'll review here.

ROM Releases

This is the most expensive medium presently available for reproducing software; however, it has utility in the convenience of use provided by built-in software. In terms of practical products, however, this form of software will most frequently show up in manufactured products preloaded at the factory, rather than user integrated software. A ROM program is difficult to achieve in a relocatable form, and has a certain permanence which is both its advantage (convenience) and its disadvantage (difficulty or impossibility of patching). This method has been successfully employed in several desk top calculator packages in the form of ROM software options, and in such personal computing products as built-in BASIC interpreters and monitors. For end user markets, this form of software can dispense with all but the user oriented manuals in most purchases, since modification is impossible.

Magnetic Digital Media Releases

As more and more floppy disk products, large and small, come to market, the use of the magnetic diskettes for software releases is becoming common. Similarly, the Philips and 3M digital tape media with standard digital recording techniques can be considered as vehicles for release of machine readable data. However, the price of the media and the costs of digitally recording and verifying each copy tend to make this method of delivery limited. It is used, quite naturally, as the vehicle for delivery of floppy disk operating systems from the manufacturers of drive interfaces, but there the writing and testing of a diskette full of data falls out of the expected quality assurance tests prior to packing and delivery. The same is true for the other forms of hardware which have related operating system software products that can be recorded.

Paper Tape

This venerable medium has been in existence longer than the modern electronic computer. No survey of distribution media would be complete without mention of it. Very reliable means exist for reading paper tapes into computers at quite economical prices, well under $100 for the peripheral. As a distribution medium, however, paper tape in my opinion suffers from several disadvantages: It is inconvenient to store, bulky, prone to create a messy tangle due to manual handling with inexpensive peripherals. Whether my opinion is supported in the marketplace is another question altogether.

Audio Media Releases

One of the most useful and practical vehicles for the distribution of software is likely to be the use of audio recording media. Here we can identify two principal methods of distribution; recording on tape cassettes or other magnetic tape audio media, and recording on the audio equivalent of a read only memory, the phonograph record.

The technology of recording on tape results in a product with a fairly high unit cost for each copy of the information. To this must be added the cost factors associated with normal printing of the rest of the documentation. Cassettes, as a recording method, are a logical choice for custom software, or small volume situations, but the high degree of manual labor associated with each copy argues against the practicality of large production runs in this form.

The technology of making phonograph records is, on the other hand, a well established mass production technique which can be adapted to the software distribution without much variation from standard methods. To illustrate the point and to test the concept, I made a test in the spring of 1976 at the suggestion of David Fylstra, an associate of mine who is also a homebrew record maker. He arranged for the cutting of a test record with the audio format of my personal monitor program, circa March 1976, to test out this method using a master record cut on standard recording industry equipment. Depending upon size and quantity of pressing, the costs per record run well under $1, which is hard to meet with the cassette duplication method.

Machine Readable Printed Media

As a final option, there is the use of machine readable printed formats for object code and relocation information in the release of software products. This is a form which was suggested to us at BYTE by Walter Banks of the University of Waterloo, and with which we have been experimenting in the pages of BYTE. In this method, an optical reader is used to scan printed materials which have been formatted into a series of bars corresponding to the digital information. Because of constraints in the design of the layout and the method of scanning, it is possible to simplify the scanner designs to the point where a very inexpensive peripheral is used together with some adaptive software which takes care of the speed tolerant input scan. The beauty of this method is that it "comes for free" in so far as actual production costs are concerned. Why is this true? The reason is that the 200 to 300 pages of documentation needed to support a systems software product with perhaps 12 K bytes of object code require only an additional five to seven pages of

Articles Policy

BYTE is continually seeking quality manuscripts written by individuals who are applying personal systems, or who have knowledge which will prove useful to our readers. Manuscripts should have double spaced typewritten texts with wide margins. Numbering sequences should be maintained separately for figures, tables, photos and listings. Figures and tables should be provided on separate sheets of paper. Photos of technical subjects should be taken with uniform lighting, sharp focus and should be supplied in the form of clear glossy black and white or color prints (if you do not have access to quality photography, items to be photographed can be shipped to us in many cases). Computer listings should be supplied using the darkest ribbons possible on new (not recycled) blank white computer forms or bond paper. Where possible, we would like authors to supply a short statement about their background and experience.

Articles which are accepted are typically acknowledged with a binder check 4 to 8 weeks after receipt. Honorariums for articles are based upon the technical quality and suitability for BYTE's readership and are typically $25 to $50 per typeset magazine page. We recommend that authors record their name and address information redundantly on materials submitted, and that a return envelope with postage be supplied in the event the article is not accepted.■

machine readable bar code copy, hardly affecting the economics of the book at all. These 200 to 300 pages of documentation are required by the product concept, whether or not there is any other form of machine readable code made available.

Economics of Publishing

With media established, and a product concept outlined, what about addressing the problem of rewarding the producers of software products? Here, as in any area of publishing, the answer is quite simple. The publishing house judges whether the particular software package is in its view a readily marketable product with a certain minimum press run potential. If so, the publisher puts up production capital, where the author puts up intellectual capital in the form of his or her work. It is a risk situation in which both parties are making a speculation that readers will purchase the product; as in numerous parallel situations throughout industry, authors and publishers work on an agreed upon split of any rewards from success in the marketplace.

Applied to the software publishing variant of this business, the author's intellectual capital is in the form of the program, its source code, its object code, and its documentation; the publisher's contribution is the marketing organization, the technical editing of the manuscripts, and the technical details of book preparation. Other than the specialized content, the method of operation and the details of the arrangement are not much different from publishing any item. Rewards to authors now become a small amount (in absolute terms) of royalty recovered from orders of magnitude in sales for successful software book products.

Proprietary Products

The problem of protecting and keeping software proprietary is no longer a major "new" issue when publishing of software is contemplated. How many people extensively copy from books? Very few, and if they attempt to make a regular practice of it they would tend to be prosecuted by publishers under copyright law. In publishing software, an implicit or explicit license to copy the copyrighted materials for personal use and modification is part of the bargain; the price is low enough so that if you want your own user documentation, you buy your own copy of the book (even if you may have been using object code derived from your neighbor's computer). Since the documentation is a necessary component of use, no sales tend to be lost in the long run due to the fact that object code can be swapped around.

Conclusions

What I have endeavored to show is that there is quite some potential for the sale and distribution of software using conventional publishing techniques with modifications to suit this type of product. By publishing software along with machine readable code, we end up with a way to make the products widely available, yet retain the desirability of compensating authors for their efforts in proportion to the success of the product.■

Circle 236 on inquiry card.

Technical Forum

More on Using the 8x300

Jon Twichell
303-D Eagle Hts
Madison WI 53705

A short note concerning your note in the March 1977 issue of BYTE (page 100) on the Signetics 8X300. This processor was designed and sold as the SMS 300 (Scientific Micro Systems, 520 Clyde Av, Mountain View CA 94043). I used SMS's development system in a scientific data acquisition system. Two years ago it was the fastest thing on the market. I was building a two dimensional multichannel analyzer out of a Modcomp II, and used the microcontroller for the address mapping and handshaking. Pure blinding speed was our object, subject to the constraint of programmability. The observation is that this is the same criterion for a microprocessor used for emulation.

True, the 8X300 is fast, but by today's standards, not that fast. It is rumored that SMS is working on an ECL version . . . . Anyway, if one examines the architecture of the 8X300, one finds two chokes, both fixable. The first is that the microcode is quite vertical, as one would expect with a 16 bit instruction. One can effectively double the speed of the system by simply extending the microcode width by eight bits. Make your program memory 24 bits wide; use 16 in the normal fashion. The extra eight are used as the address in working storage or IO space. The 8X300 must use an instruction to load the "memory address register" (IVR REG) and then another instruction to fetch the word at that address; by selecting an address for each instruction (the extra eight bits), most programs halve in size, and double in speed.

The second choke is the time multiplexed IO bus. I seriously suggest the user extend the microcode and stay away from those

Continued on page 110

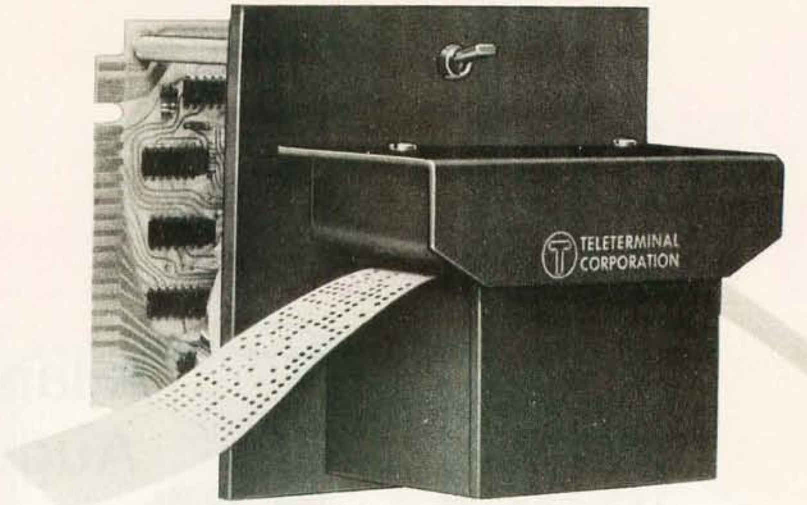


Photo 1: The Teleterminal Corporation Fly Reader for use with the KIM-1 microprocessor.

Come Fly With KIM

Rick Simpson
MOS Technology Inc
Valley Forge Corporate Center
950 Rittenhouse Rd
Norristown PA 19401

Many computer hobbyists start with nothing more than a processor, a small amount of programmable memory, a small onboard monitor such as MIKBUG or KIM and some front panel switches. Those with more foresight, or cash, will have a keypad or even a full keyboard for data entry and processor control. But even with a good monitor and a full keyboard and display, loading programs is a tedious chore at best, and there is an awful feeling when you turn off power, knowing that twenty minutes of typing just evaporated.

The next step in expanding the system is usually an audio cassette interface or a Teletype with paper tape reader and punch for the wealthy or fortunate. Now the tedious retyping is eliminated and a program, once written and recorded or punched, can be reloaded in a matter of minutes.

Many people stop at this point. When hand assembly of programs is required, a program of more than a few hundred bytes is rarely attempted. But as the software gap is slowly filled, more and more systems are being implemented with assemblers or BASIC interpreters. More memory is purchased to expand programmable memory from a few hundred bytes to 4 K, 8 K, or more. *[One firm now even markets a 64 K board!. . .* CH*]* Once again your memory has outrun your ability to fill it in a reasonable time.

For instance, using the Teletype paper tape reader or audio cassette interface on the KIM system, a 2 K Tiny BASIC interpreter takes almost ten minutes to load. A 12 K BASIC source program would take an hour. Even a 30 character per second interface only cuts this to twenty minutes. The alternatives seem to be a Tarbell or Suding type high speed cassette system, a 3M drive at 9600 bps or a floppy disk.

The floppy disk certainly solves the speed problem. We can now load 12 K in a few seconds, but at a cost of $1,000 to $2,000. The high speed cassette is reasonable in cost, about $200 including the high quality cassette unit required, but tricky to interface unless a manufacturer-supplied board or kit is available.

Although some magnetic tape units have start, stop, and search functions under program control, most users end up pushing the buttons. No hobbyist magnetic tape cassette unit can read a few (ie: one line or so) characters, stop and process the data, and then start and read some more, a real need when running an assembler with the source stored on the tape in a limited resource system.

After this lengthy preamble, you may have suspected that I have an alternative solution in mind, and I do: a high speed paper tape reader, manufactured by the Teleterminal Corporation, called the Fly Reader, shown in photo 1. Although a bit more expensive than the high speed cassette system (about $350), it is far faster; reading at 300 characters per second, it can load my Tiny BASIC in twenty seconds, or fill that 12 K of memory in two minutes. It is easy to interface, requires little software, and is extremely reliable. It needs only a single +5 V, 2 A power supply and is operated completely under program control. You can read as little as a single character at a time and can read in either direction; try that on your cassette!

Paper tape has always been the standard mass storage device for minicomputers, until floppy disks came along, and paper tape has been the most universal and inexpensive method of software distribution and interchange in the minicomputer field.

The basic problem is that it is only a reader; how do you punch the tape? There are several answers: Flexowriters and other similar low speed punches are becoming available, as are gobs of older 7 level machines. I've also seen higher speed punches, typically 60 characters per second, advertised for under $100. The fact that the punch is slow is not so important; typically you punch a tape once and read it many times. Even if you have no punch, the reader is a useful peripheral because much software is available already punched.

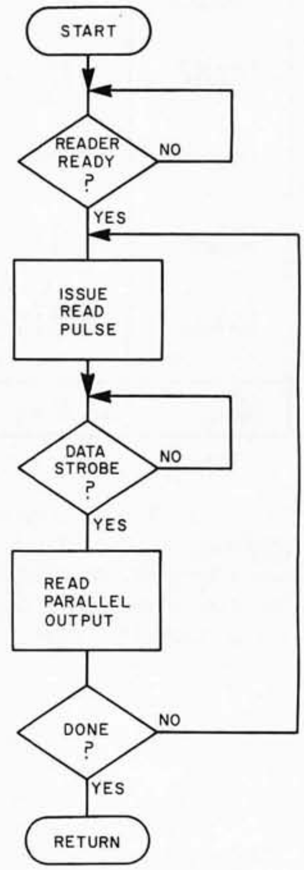
How it Works

The Fly Reader can read at such a high speed because it transfers 8 bits in parallel and contains only a single moving part: a stepping motor connected to a toothed wheel which engages the sprocket holes in the paper tape. Sensing of the holes in the tape is done by photodetectors rather than the mechanical fingers used in a low speed reader. This is a method similar to that used in the manual reader sold by Oliver Audio. Figure 1 shows a block diagram of the unit.

There are five control lines for the unit. All are compatible with standard TTL circuitry. The "load status" line is a logic 1, +5 V, if the reader is not ready because the feed gate is not closed. When tape is inserted and the gate is closed, this signal goes to logic zero.

In operation, the reader must be checked by software to see if the "reader ready" signal is at logic 1 to indicate that the reader is ready to read another tape character. The software must then issue a pulse from logic 1 to logic 0 whose width is between 500 ns and 500 μs. This READ pulse will start the reader and drive the reader ready line to logic 0. The software then watches the "data strobe" line. When data strobe goes to logic 1, the data can be read from the eight parallel output lines. If the program needs to read another character from tape, it must wait until reader ready goes back to logic 1, issue another read pulse, and wait for another data strobe. Figure 2 shows the flowchart for such software and figure 3 shows the interface timing diagrams.

Figure 2: Flowchart of the software for reading the paper tape with the Fly Reader.



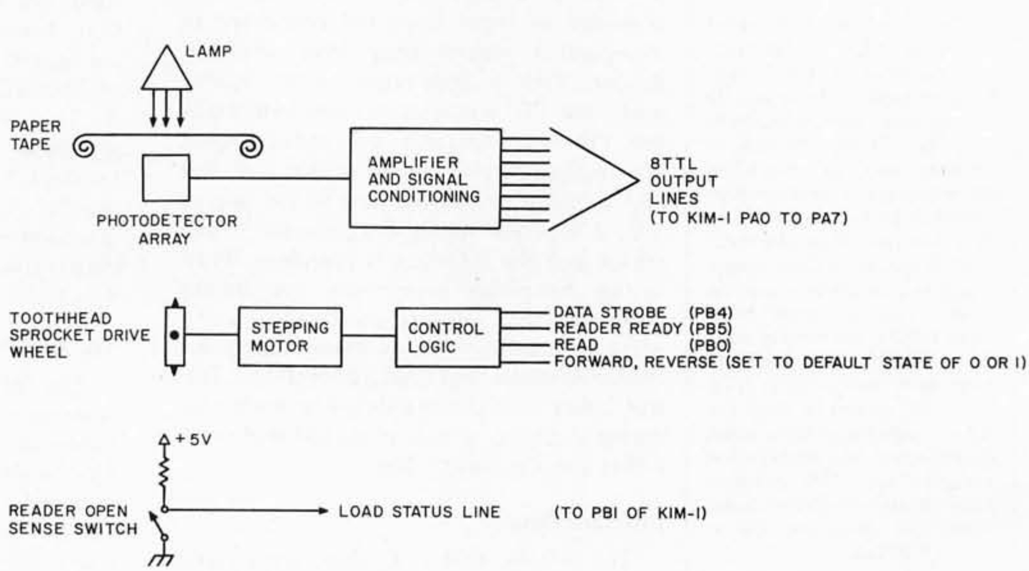


Figure 1: Block diagram of the Fly Reader. The input is achieved through an incandescent lamp and a photodetector array. The tape is advanced by a stepping motor allowing input of data either forwards or backwards. The reader open sense switch is closed when the paper tape is in the reader.

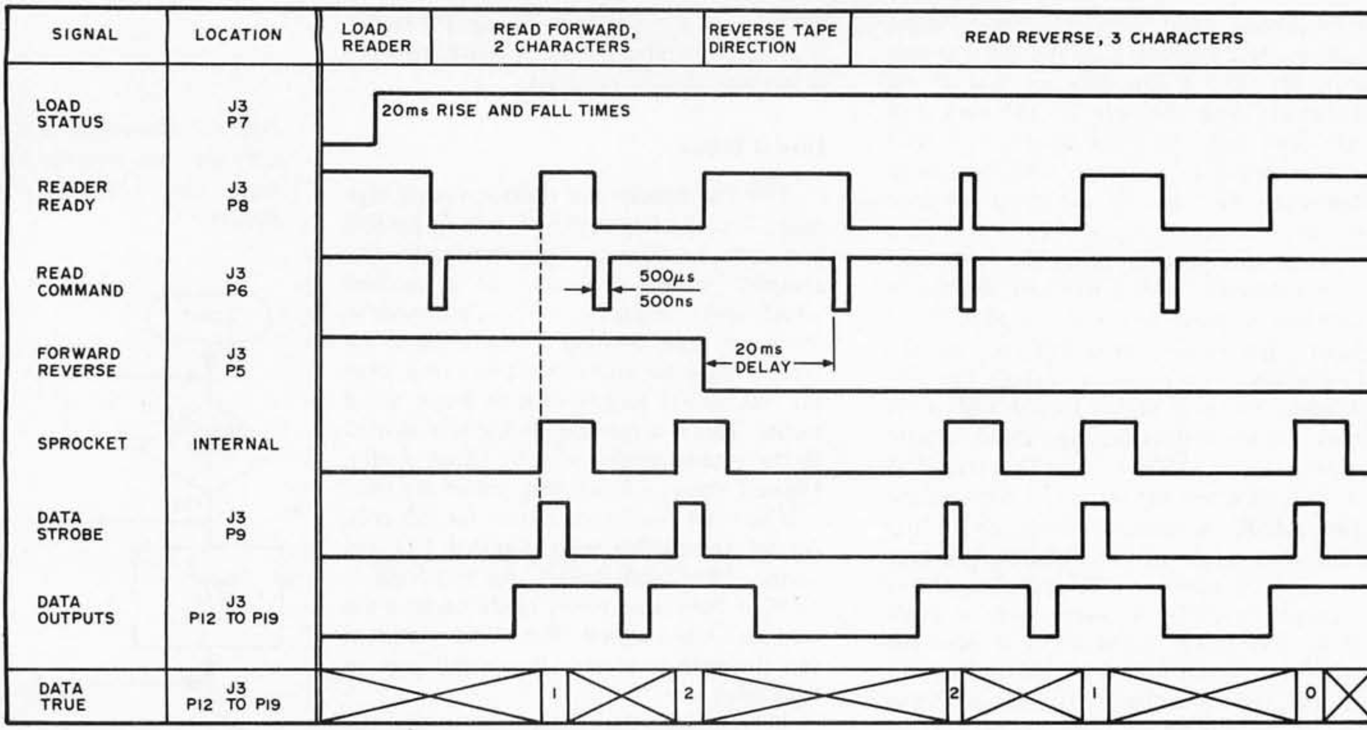


Figure 3: Timing diagram generated by the software of listing 1. The minimum width of the data strobe is 50 µs except when forced low by a new read command. The crossed out sections in the data true section indicate that the state of the output is unknown. A read command is issued only when the reader ready line is high.

Interfacing to Kim

As described above, the Fly Reader interface requires eight parallel input lines, three input control lines and one or two output control lines. Two output control lines are needed if the forward, reverse function is used; otherwise only one control line is needed. Since KIM-1 has 15 bidirectional IO lines the interface is very simple. The A data port lines PA0 to PA7 are programmed as input lines and connected to the parallel output lines from the Fly Reader. PB5 is connected to the reader ready line, PB1 is connected to the load status line, PB0 is connected to the read command line and programmed as an output line, and PB4 is connected to the data strobe line. A 5 V, 2 A power supply is connected to the reader and the interface is complete. When wiring the power connectors, you should make sure that separate power and ground wires are run back to the power supply for both the motor and logic connections. This will insure that current surges to the motor during stepping operations do not feed noise pulses into the control logic.

Interface Details

The fifteen KIM-1 IO lines are divided into two ports. Each port has a data direction register and a data register. Writing a 1 to a bit or bits in the data direction register configures the corresponding IO lines for output, writing a 0 sets them for input. For instance, writing a hexadecimal 02 to the A data direction register configures the PA0 line for input, PA1 for output, and PA2 through PA7 as input lines. Similarly, writing hexadecimal F0 to the B data direction register configures PB0 through PB3 as input lines and PB4, PB5 and PB7 as output lines. Note that there is no PB6, and PB7 has no output pullup; it is essentially an open collector output. Reading the A or B data register will show whether the signal at each input line is 1 or 0 and will show whether a 0 or 1 was previously written to any lines configured as outputs. Writing to a data register will set the appropriate output lines to 1 or 0 and does not affect lines programmed as inputs. Hexadecimal address location 1700 is the A port data direction register, hexadecimal 1702 is the B data direction register, hexadecimal 1701 is the A data register, and hexadecimal 1703 is the B data register.

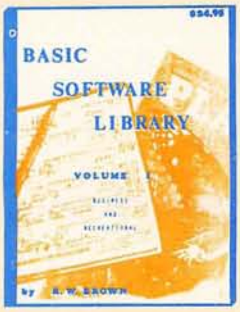
The KIM Paper Tape Format

The software to drive the reader uses the same paper tape format as that used in the KIM-1 Q (paper tape dump) and L (paper tape load) commands. Thus any paper tape punched on a low speed punch by KIM-1 can be read by the Fly Reader. The KIM-1 paper tape format ignores any characters

Fantastic Software

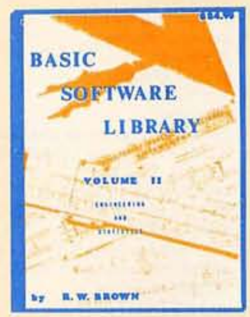
This LIBRARY is a complete do it yourself kit. Knowledge of programming not required. EASY to read and USE

Written in compatible BASIC immediately executable in ANY computer with at least 4K, NO other peripherals needed.



This Library is the most comprehensive work of its kind to date. There are other software books on the market but they are dedicated to computer games. The intention of this work is to allow the average individual the capability to easily perform useful and productive tasks with a computer. All of the programs contained within this Library have been thoroughly tested and executed on several systems. Included with each program is a description of the program, a list of potential users, instructions for execution and possible limitations that may arise when running it on various systems. Listed in the limitation section is the amount of memory that is required to store and execute the program.

Each program's source code is listed in full detail. These source code listings are not reduced in size but are shown full size for increased readability. Almost every program is self instructing and prompts the user with all required running data. Immediately following the source code listing for most of the programs is a sample executed run of the program.



The entire Library is 1100 pages long, chocked full of program source code, instructions, conversions, memory requirements, examples and much more. ALL are written in compatible BASIC executable in 4K MITS, SPHERE, IMS, SWTPC, PDP, etc. BASIC compilers available for 8080 & 6800 under $10 elsewhere.

This Library is destined to become one of the reference bibles for the small computer field, due to its versatility and uniqueness and the ease of operation of the programs it contains. These volumes are deductible as a business expense when purchased by a company. Send your remittance for prompt delivery, while supplies last. Volume discounts are available to qualified dealers.

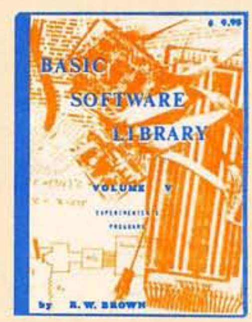
VOLUME I
Bookkeeping
Games
Pictures $24.95

VOLUME II
Math & Engineering
Plotting & Stat
Basic Statement Def $24.95

VOLUME III
Advanced Business
Billing
Inventory
Payroll $39.95

VOLUME IV
General Purpose $9.95

VOLUME V
Experimenter $9.95



FUTURE ADDITION TO THE "BASIC SOFTWARE LIBRARY"

Volume VI (A Complete Business System $49.95) General Ledger System - Taxes, Pyrl, W-2's, Inventory, Depr., Financial Statements, etc. AVAILABLE MID SUMMER

Available at most computer stores.
Add $1.50 per volume for postage and handling.

SRI

SCIENTIFIC RESEARCH

1712-B FARMINGTON COURT
CROFTON MD 21114

Phone Orders call (800) 638-9194
Information and Maryland Residents Call (301) 721-1148

Circle 282 on inquiry card.

address	hexadecimal code	label	op.	operand	commentary
1C4F		START	EQU		$1C4F
4000	D8		CLD		clear decimal mode;
4001	20 57 40	PTRLD	JSR	PTRINI	go to PTRINI;
4004	20 6F 40	LOAD	JSR	GETPTR	go to GETPTR;
4007	C9 3B		CMP	$3B	if A not equal to 3B go to LOAD;
4009	D0 F9		BNE	$LOAD	
400B	A9 00	LOADS	LDA	00	else A:=00;
400D	85 F7		STA	$F7	store checksum;
400F	85 F6		STA	$F6	
4011	20 8B 40		JSR	PTRBYT	go to PTRBYT; [get byte count]
4014	AA		TAX		X:=A;
4015	20 91 1F		JSR	$1F91	compute checksum;
4018	20 8B 40		JSR	PTRBYT	get high address;
401B	85 FB		STA	FB	store high address pointer;
401D	20 91 1F		JSR	$1F91	compute checksum;
4020	20 8B 40		JSR	PTRBYT	get low address pointer;
4023	85 FA		STA	FA	store low address pointer;
4025	20 91 1F		JSR	$1F91	compute checksum;
4028	8A		TXA		A:=X;
4029	F0 0F		BEQ	LOAD3	if A:=0 go to LOAD3;
402B	20 8B 40	LOAD2	JSR	PTRBYT	get data;
402E	91 FA		STA	FA,Y	store data;
4030	20 91 1F		JSR	$1F91	compute checksum;
4033	20 63 1F		JSR	$1F63	get next address;
4036	CA		DEX		X:=X–1;
4037	D0 F2		BNE	LOAD2	go to LOAD2;
4039	E8		INX		X:=X+1;
403A	20 8B 40	LOAD3	JSR	PTRBYT	get data;
403D	C5 F6		CMP	$F6	compare high order checksum;
403F	D0 12		BNE	LOADER	if different go to LOADER;
4041	20 8B 40		JSR	PTRBYT	else get data;
4044	C5 F7		CMP	$F7	compare checksum;
4046	D0 0B		BNE	LOADER	if different go to LOADER;
4048	8A		TXA		else A:=X;
4049	D0 B9		BNE	LOAD	if A not equal to 0 go to LOAD;
404B	A2 0C	LOAD7	LDX	0C	else X:= location of 'KIM';
404D	20 31 1E	LOAD8	JSR	$1E31	output message;
4050	4C 4F 1C		JMP	START	go to START;
4053	A2 11	LOADER	LDX	11	X:=location of 'ERR KIM';
4055	D0 F6		BNE	LOAD8	go to LOAD8;
4057	A9 01	PTRINI	LDA	$01	[initialization routine]
4059	8D 03 17		STA	$1703	A:=B port address;
405C	8D 02 17		STA	$1702	read flag:=1;
405F	AD 02 17		LDA	$1702	A:=B register;
4062	29 02		AND	$02	determine PB1;
4064	D0 08		BNE	OK	if reader ready go to OK;
4066	A9 58		LDA	'X'	else A:= 'X';
4068	20 A0 1E		JSR	$1EA0	output 'X';
406B	4C 57 40		JMP	PTRINI	go to PTRINI;
406E	60	OK	RTS		return;
406F	AD 02 17	GETPTR	LDA	1702	[subroutine to input one character]
4072	29 20		AND	$20	get bit from B data register;
4074	F0 F9		BEQ	GETPTR	if not ready go to GETPTR;
4076	A9 00		LDA	$00	else output read pulse;
4078	8D 02 17		STA	$1702	
407B	A9 01		LDA	$01	turn off read pulse;
407D	8D 02 17		STA	$1702	
4080	AD 02 17	CHECK	LDA	$1702	get bit 5 from B data register;
4083	29 10		AND	$10	
4085	F0 F9		BEQ	CHECK	if character not ready go to CHECK;
4087	AD 00 17		LDA	$1700	else get character;
408A	60		RTS		return;
408B	20 6F 40	PTRBYT	JSR	GETPTR	get character;
408E	20 AC 1F		JSR	$1FAC	pack character;
4091	20 6F 40		JSR	GETPTR	get another character;
4094	20 AC 1F		JSR	$1FAC	pack character;
4097	A5 F8		LDA	$F8	A:=2 characters;
4099	60		RTS		return;
409A	20 57 40	MAIN	JSR	PTRINI	go to PTRINI;
409D	20 6F 40	LOOP	JSR	GETPTR	go to GETPTR;
40A0	4C 9D 40		JMP	LOOP	go to LOOP;
			END		

Listing 1: The basic software needed to run the Fly Reader with the KIM-1 microprocessor. The software uses the KIM-1 monitor routines and substitutes the GETCH and GETBYT routines with routines GETPTR and PTRBYT. The new subroutine PTRINI properly configures the IO lines used with the reader. Subroutine PTRINI will output an endless number of 'X' characters until the tape is loaded into the reader. The listing was set prepared from a cross-assembly provided by the author. A symbol table showing the values of the symbols used and where they are referenced follows the assembly and will prove useful when it is necessary to relocate a program at a different starting address.

CROSS REFERENCE TABLE

Symbol	Value	Referenced
CHECK	4080	4085
GETPTR	406F	4004 4074 408B 4091 409D
LOAD	4004	4009 4049
LOADER	4053	403F 4046
LOADS	400B	****
LOAD2	402B	4037
LOAD3	403A	4029
LOAD7	404B	****
LOAD8	404D	4055
LOOP	409D	40A0
MAIN	409A	****
OK	406E	4064
PTRBYT	408B	4011 4018 4020 402B 403A 4041
PTRINI	4057	4001 406B 409A
PTRLD	4001	****
START	1C4F	4050

read until a semicolon is found; the next two characters give the hexadecimal number of bytes on the current line to be punched. This is followed by four characters, two bytes, giving the high order and low order bytes of the starting address for the data to follow. This is followed by the data which KIM-1 software always punches 24 bytes per line, a 2 character checksum for the line, and a carriage return. The carriage return is followed by six null characters, and a semicolon starts the next line. The last line punched contains 0 for the number of bytes, 0 for the address bytes and is followed by a four character checksum. When finished reading a paper tape, KIM-1 types 'ERR KIM' if the checksum does not compute (there has been an error in reading the tape), or just 'KIM' if the tape was read correctly.

The software consists of a copy of the KIM-1 monitor routine for reading paper tape modified by removing all calls to the GETCH and GETBYT routines and substituting two new routines, GETPTR and PTRBYT. A new subroutine, PTRINI is called at the beginning of the mainline program to properly configure the IO lines and check that the read head on the Fly Reader is closed. If it is not, KIM-1 will type out the character 'X' endlessly until the read head is closed. The software shown in listing 1 occupies 154 bytes starting at hexadecimal location 4000.

The Fly Reader is an excellent way to add a high speed paper tape reader to a microprocessor system. It is easy to interface and requires only a single +5 V supply. As a fast paper tape device it is considerably faster than an audio tape cassette system and offers increased flexibility of operation.■

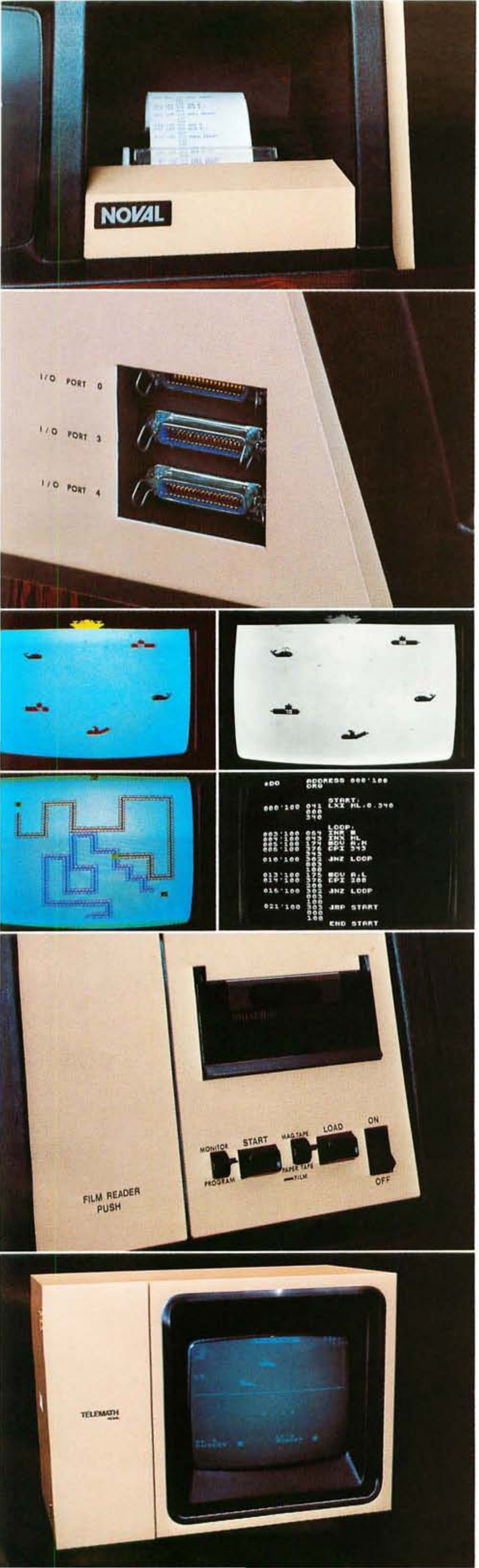
THE HOME COMPUTER COMES OF AGE

Beyond state-of-the-art capabilities, handsomely housed in a fine piece of functional furniture.

Introducing the NOVAL 760 Computer

Noval Incorporated proudly introduces the finest personal computer ever offered, the NOVAL 760 COMPUTER. This fully computational, self-contained hardware and software package offers limitless expansion capabilities. NOVAL's extensive and expanding library of programs presents exclusive opportunities for educational applications, practical implementations and a world of fun. And you can develop low-cost end user products never before available at the small board level.

The NOVAL 760 COMPUTER, in its handsome desk setting will take you wherever your imagination wants to go.



The Ultimate in Home Computers

$2995 includes a fully-assembled, fully-tested personal computer with:

- Handsome wood desk, designed to complement any decor
- Crisp black and white 12" professional display monitor
- Full, easy-to-use Alpha-Numeric Keyboard
- Printer
- Extensive graphic capability
- Fully remote-control, professional mag tape system
- 8080 Microprocessor System
  - 16K RAM
  - 3K PROM
- Display control, with additional
  - 1K Refresh RAM
  - 2K Character generator RAM
  - 1K Scratch pad RAM
- 8 built-in I/O ports
- Power Supply
- System Software on mag tape, with
  - Interactive Editor/Assembler
  - I/O control for
    - Printer
    - Display (black & white or optional color)
    - Mag Tape Unit
    - Audio Tone Generator
    - Keyboard
    - Film Reader
    - Paper Tape Reader
  - Graphics
  - On-line debugging technique
  - E-PROM burner
  - Copy-verify
- Comprehensive and complete Operating Manual *(also available separately)*

The NOVAL 760 COMPUTER Instant-Edit feature makes Assembly Language interactive AS YOU WRITE IT! The computer corrects errors before they're entered.

And Look at these Additional NOVAL Capabilities:

- System Software on PROM
- BASIC on Mag Tape
- BASIC on PROM
- Bright color display screen which converts for TV use
- Second display screen
- 16K additional RAM
- E-PROM Burner
- Bus-extender
- Film reader
- Paper tape reader
- Dual hand-held keyboards for competitive game action and, for the first time on a home computer,
- Basic Graphics!

The NOVAL 760 COMPUTER Operating Manual

This comprehensive, yet easy-to-understand, Noval Operator's Manual provides an exciting introduction to the capabilities and possibilities of microprocessor technology. In step-by-step, "how to" fashion, the intriguing inner-workings of the NOVAL 760 COMPUTER are revealed (included with your 760 Computer or available separately for $20.00).

Challenging Libraries Available Now

Add the excitement of action-packed competition as your child learns the elements of math with Noval's extensive TELEMATH library. TELEMATH, a computer audio-video-graphic instructional system developed by Noval, presents challenging math problems to each of 2 players. The first player responding correctly moves one step closer to victory in this graphic game format.

Our exclusive affiliation with Gremlin Industries, the leader in commercial computer videographic and wall games, offers you the opportunity to play some of the most sophisticated games ever developed at home! With the Gremlin library, your NOVAL 760 COMPUTER becomes a home entertainment center.

The expanding libraries of TELEMATH and Gremlin are available to all NOVAL 760 COMPUTER owners.

The NOVAL 760 COMPUTER. A fully-assembled, fully tested personal computer . . . not a kit!

If you can imagine it, you can achieve it with the NOVAL 760 COMPUTER.

Classified Ads

FOR SALE OR TRADE: Issues 1, 2 and 3 of BYTE for $10 or exchange for May 1976 issue and $5. CPT William T Pace, 5433B Seay St, Fort Polk LA 71459. (318) 537-7198.

FOR SALE: *Computers in Laboratory Medicine.* I have several copies of this book which are surplus to a class requirement. This excellent book is edited by Derek Enlander MD and is one of the most up to date and forward looking works on the subject of computerization in nuclear medicine, clinical pathology, anatomic pathology and diagnostic data retrieval. The book was published by Academic Press in 1975-1976 and costs $14. David Johnson, 1473 Pine St, San Francisco CA 94109.

FOR SALE: IBM 2311 compatible disk drive made by Marshall, plus disk pack, $425. Teletype ASR 33 with auto-answer modem, auto-dial, stand. Three MITS 4 K dynamic boards, operating. Signetics 3000 Microprocessor Designer's Kit. Make offer. Lenny Heath, Bob Turnage, 86 Village Grn, Greenville NC 27834. (919) 752-7813.

FOR SALE: Two MITS 4MCD memories, one S-4 K update kit, less RAMS. All assembled, all nonfunctional. First cashiers check for $145 takes it all. K K Tatlow, 303 S 4th St, Rockford IL 61108.

FOR SALE: Friden Flexowriter Model 2303 with paper tape punch reader, 7 level, types 100 wpm in upper and lower case, works good, recently serviced, can be converted to ASCII, $150 plus shipping. J E Upchurch, POB 1987, Sebring FL 33870. (813) 385-2788.

FOR SALE OR TRADE: One Cartrivision video tape recorder, camera and tapes, $650 cash or Altair, IMSAI equivalent with memory. Doug Thurston, POB 1104, Silsbee TX 77656. (713) 246-3091.

FOR SALE: Altair 8800 computer with 2 MITS 4 K Dynamic memory cards, 88-ACR cassette interface, 88-2SIO serial board, 88-PIO parallel board, 2 expander boards, 8 K BASIC on cassette. Not completely assembled. Original cost was over $1200. Will sell for $750. Bob Majdanski, 214 Coolidge Av, Hasbrouck Heights NJ 07604. (201) 288-3742 after 7 PM.

WANTED: BYTE numbers 1, 2, 3, 8, 9 and 10. I will consider any offer, even partial ones. (One issue is better than none.) Send your terms and asking price to me and I'll get in touch with you. Tony Aiuto, 19 Old Field Ln, Lake Success NY 11020.

FOR SALE: Sphere 340 system with 20 K programmable memory plus 16 K memory board less memory chips, 1 K EPROM system monitor, full SIM board (one cassette and modem), 80 column line printer with tractor feed, dual floppy disk, 9 inch CRT in console, large power supply, 4 K EPROM board (1702A), no EPROM chips. Software includes assembler, editor, IO handler, memory test and disassembler. This system is ideal for program development. Working perfectly, $6500. Wayne Smith, 227 S State St, Salt Lake City UT 84111. (801) 363-4941.

FOR SALE OR SWAP: One Mohawk Data Systems 1101 keyboard and tape unit. It's in good working condition with loads of electronics and it would even make a nice desk! Will trade for Altair 8800 interface, memory, terminal or best offer of cash. Write G Ryan, Rm 1-15, Off Campus Apts, Potsdam NY 13676.

FOR SALE: HP-65, including a case, charger, manuals, programming forms, over 100 magnetic cards with tested prerecorded programs, and 40 blank magnetic cards for your new programs. Any of over 140 HP-65 Users' Library programs available for copying cost. Remember, the HP-65 is faster than the HP-67; so is better for problems where speed is needed. $300. Delmer D Hinrichs, 2116 SE 377th Av, Washougal WA 98671.

FOR SALE: IMSAI 8080 with 22 card mother board, new in box, $645. Also, will assemble and debug it for you for $120 extra. S Kim, 10190 Vicksburg Dr, Cupertino CA 95014. (408) 996-0537, after 7 PM.

FOR SALE: Two PDP-12s (PDP-8 and Line); four 4 K 12 bit memory units; three DF 32 disks, 32000 words 16.7 μs access time; two PC-12 high speed tape readers, 300 cps; two PC-12 high speed tape punches, 50 cps; two AFO1-AMO8 analog converter systems; two IO bus converters; two Teletypes; Dual-MS and OS-8 software and a lot of Decus software for the PDP-8, BASIC, FORTRAN and 8 K assemblers. All this for $5,500. I'll sell the units as two or three PDP-12s separately or together. Contact Keith Elkin, Dianavagen 30, 115 43 Stockholm SWEDEN 08/67 35 68.

FOR SALE: Model 15 Page Printer – Baudot Teletype suitable to OSI, MITS 680b computers. Excellent working condition, with current source and box of paper. Also, Friden Auxiliary Punch – Baudot paper tape punch. Could be modified to ASCII with additional parts, not supplied. Probably works, not guaranteed, with stand and motorized tape take up spool. Make offers for either to G A R Trollope, 433 Cherry Ln, Lewiston NY 14092. (716) 754-7222.

FOR SALE: M6800 cross assembler. A two pass cross assembler written in FORTRAN IV is available for the M6800 Motorola micro. Input is in fixed format. Statements are similar to Motorola assembler language, most features of the language being supported. Additionally, a system symbol table is supported, enabling symbolic reference to system addresses and assembly of routines to contiguous memory locations. Send $1 for the manual, and $5 for the listing or $8 for a paper tape (state if XOFF needed) to G A R Trollope, 433 Cherry Ln, Lewiston NY 14092. (716) 754-7222.

FOR SALE: CDP-1802 microcomputer software, Morse keyboard, 256 bytes, $3. WA6UYV, 4956 Andrea Blvd, Sacramento CA 95842.

FOR SALE: Viatron System 21 in working condition. Must sell to make room for 8080 system, $300; you pay shipping. Jim Williams, 4175 Walnut Ridge Dr, Columbus OH 43224, (614) 889-3836, (days).

TIMESHARING SERVICES WANTED: Research and advanced degree level programs and routines of civil and mechanical engineering, or bio-engineering or bio-medical interest. Please send full information and price list to Donald Becnel, POB 14473, Baton Rouge LA 70808.

WANTED: Someone who has or wants to write an 8 K version of BASIC for my COSMAC ELF, as in the *Popular Electronics* August 1976 construction article. It uses the RCA CDP 1802 central processor unit. Please send any and all correspondence on this subject to Greg Watson, 9617 Jomar Dr, Fairfax VA 22030.

FOR SALE OR TRADE: BYTE issues #1 thru #10, $50, or swap for first six issues of *73* magazine or first three issues of *CQ* magazine. Wanted: Flexowriter with upper and lower case type; will buy or swap for three upper case only machines, one designed for computer interface. Greggory S Walls, 5630 Ensenada Way, Riverside CA 92504.

TELEPRINTER: Creed 7B. Serial input/output device. Swap for KIM-1 or similar microprocessor. Free delivery. All BYTE back issues, 1975 and 1976 (excluding numbers 3 and 10), included. T N Arthur, 30 Willingdon Av, Kloof 3600, Natal Republic of SOUTH AFRICA.

FOR SALE: Vidar #624 computer clock, 999 days, 23:59 hours, 23:59 minutes and 60 seconds, BCD output (like new), $200. Also, 1 Brpe (used) $100. FOB, Norwich NY 13815, (607) 334-4478.

WANTED: BYTE issues, October and November 1975, (#2, 3). For sale or prefer to trade for the above issues, December 1975, June thru December 1976, (#4, 10, 11, 12, 13, 14, 15 and 16). David A Eastman, 46 Sable Av, N Dartmouth MA 02747, (617) 993-7098.

FOR SALE: BYTE #1 in mint condition (never removed from mailing wrapper). Best offer over $20. Steve Saunders, 5400 Ellsworth Av, Pittsburgh PA 15232.

FOR SALE: MCS 6502 resident assembler. Uses 6530 TIM input and output routines. Can be easily modified to be used in systems not having the TIM chip. Resides in less than 2 K memory (4000-47FF). This software contains a text editor (TED) which sets up the source file, and a 2 pass assembler which assembles the source and generates a symbol table. If an error occurs, an error code and line number are outputted which pinpoints the error in the source file. Source file and symbol table can be located practically anywhere in memory. Assembled programs can be executed via the TED R (Run) command. Hexadecimal listing and operator's manual, $15. Source code and manual, $25. C W Moser, 3239 Linda Dr, Winston-Salem NC 27106.

FOR SALE: M6800 cross assembler written in FORTRAN and set up to run under any system with some minor modifications. Other M6800 programs available. Mid-Michigan Computer Club, T Preston, 15151 Ripple Dr, Linden MI 48451.

KIM-1 USERS: Am designing a universal PC board same size as KIM-1 board. Double sided, space for up to 70 16 pin and 12 28 pin chips, four regulator or power transistor packages with heat sinks. Need ten people to get cost to $30 each. Contact Gerry Houlder, 3832 Stevens Av, Minneapolis MN 55409.

WANTED: The base for the CRT terminal enclosure that was offered by Tri-Tek. Will buy outright or swap for TTL chips or back issues of BYTE. If you have one of these bases for sale, write Larry Ingram, 20 Locust St, Cambridge MA 02138.

AUCTION: Issues 1, 2 and 13 of BYTE. Good condition. Make your offer of cash or equipment for use with Altair 8800. Bidding closes last day of month this is published. Enclose SASE and you will be notified if you are not a successful bidder. Also for sale: DEC PDP-11 regulators 5 V 25 A, $30; 15 V 10 A, $20. H S Corbin, 11704 Ibsen Dr, Rockville MD 20852.

FOR SALE: DEC PDP-8E with following options: MM8E EJ 8 K core memory, KM8E memory extension and timeshare control, KC8E programmers console, KP8E power fail detect, KL8E console Teletype control, BE8A omnibus expander. Asking $3000. If interested contact David Kohl, 3002 Bedford Av, Brooklyn NY 11210. (212) 952-4420 days.

COMPUTER Paper tape punches and readers, 8 and 5 level. IBM 056 Verifier, ultra high speed paper tape punch (2400 ppm), 5 level. PDP-8e/Lab8e software to exchange. 150 item list of ham, computer, test, hi fi, aircraft, photo, and antique gear for sale or trade. SASE. K2DCY, 11 Squire Hill Rd, N Caldwell NJ 07006.

FOR SALE: Teletype Model 33 KSR, $250. Model 33 ASR, $350. Friden Programatic Flexowriter Model SPD with tape punch and tape reader, $225. Friden Programatic Flexowriter Model 2201 with tape punch and tape reader, $325. Send SASE for small list. Ron Turnure, 206 S Highwood Av, Glen Rock NJ 07452.

Readers who have equipment, software or other items to buy, sell or swap should send in a clearly typed notice to that effect. To be considered for publication, an advertisement should be clearly noncommercial, typed double spaced on plain white paper, and include complete name and address information. These notices are free of charge and will be printed one time only on a space available basis. Insertions should be limited to 100 words or less. Notices can be accepted from individuals or bona fide computer users clubs only. We can engage in no correspondence on these and your confirmation of placement is appearance in an issue of BYTE.

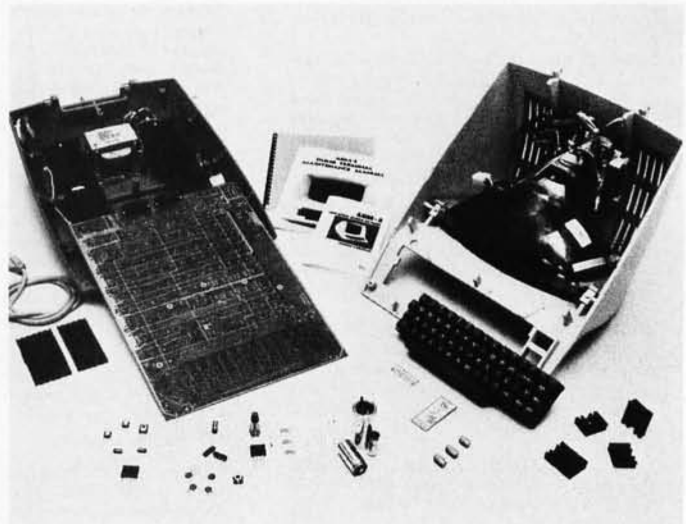
Please note that it may take three or four months for an ad to appear in the magazine.■

synchro-sound enterprises

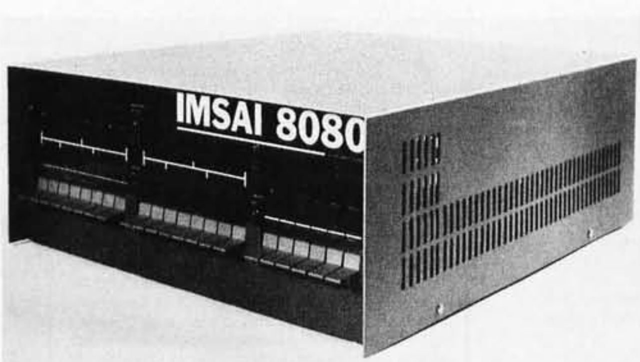
"THE COMPUTER PEOPLE"

SPECIAL $839.95

" IN STOCK "
NEW LEAR SIEGLER ADM - 3A KIT
FULL ADDRESSABLE CURSOR



CHARACTER GENERATION
5 x 7 dot matrix.
DISPLAY FORMAT
Standard: 1920 characters, displayed in 24 lines of 80 characters per line.
CHARACTER SET
Standard: 64 ASCII characters, displayed as upper case, plus punctuation and control.
COMMUNICATIONS RATES
75, 110, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19,200 baud (switch selectable).



IMSAI 8080 MICROCOMPUTER
POWERFUL • EASY TO USE • LOW COST

$619.95/22 SLOT MOTHER BOARD
849.95/WITH Z-80 CPU

COMPUTER INTERFACES
EIA standard RS232C and 20 mA current-loop (switch selectable).
DATA ENTRY
New data enters on bottom line of screen; line feed causes upward scrolling of entire display with top-of-page overflow. Automatic new line switch selectable, end-of-line audible tone.

ADM-3A KIT .... $ 839.95
ADM-3A ASSEMBLED 1099.95
LOWER CASE OPTION 89.00
(when ADM-3 kit is purchased from us .... 59.95)

FOLLOWING MANUFACTURERS IN STOCK PERIPHERAL VISION, iCOM, TDL, OAE, PROCESSOR TECH., SWTP, APPLE, HAYDEN, TARBELL, IMSAI, LEAR SIEGLER, OKIDATA, DEC, JAVELIN, TELETYPE ASR-33, TRW.

SPECIAL 9" JAVELIN HIGH RESOLUTION VIDEO MONITOR $159.95

* * * * * * * * * * * * * * * * *
* 2708 EPROM $59.95 *
* Set of seven $350.00 *
* * * * * * * * * * * * * * * * *

Circle 96 on inquiry card.

Software for the Economy Floppy Disk

Dr Kenneth B Welles
General Electric, Nela Park
2623 Fenwick Rd
University Heights OH 44118

The two fundamental routines needed for a floppy disk system are: Write a block of data to the disk, and read a block of data from the disk.

As hobbyists are rapidly finding out, even the most sophisticated hardware is next to useless without the proper software to control it. My previous article on the floppy disk drive interface (see February 1977 BYTE, page 34) described a hardware device of the simplest and, consequently, most software dependent type. This month's article describes the operation and use of the routines needed for transferring data between the computer and a disk drive (one of up to eight) connected to the interface.

The two fundamental routines needed for a floppy disk drive system are: Write a block of data to the disk, and read a block of data from the disk. This sounds simple in theory but in practice much more information and many operations are needed. How many bytes of data are in the block to be transferred, and which disk drive should the block go to or come from? At which track and sector is the block to be located? How will an error be detected (and if detected what steps should be taken for correction)? These are some of the major questions involved, without even considering such specific details as data format, file structure, unrecoverable errors, directory structures, naming and dating conventions, and so on ad futilitum.

A block read or write routine can be divided into four stages:

- Set up for the data transfer
- Transfer the actual data to or from the disk
- Error detection
- Error correction

These stages must, by their definition, occur in the order listed, and in most disk systems all four stages are included. In some operating systems the error detection and correction stages are ignored during a write operation. While this speeds up transfers by eliminating a reread or verify operation, it means that most write errors will be unrecoverable.

Write Set Up

Because this interface is quite unsophisticated, data to be written onto the disk must be prepared in memory in exactly the manner that it will appear on the disk. A preamble containing 16 bytes of zeroes (128 "0" bits) and a byte boundary synchronization signal (or sync byte) must precede the data. Before the data is written, the error detection bytes must be calculated and stored with the data to be written, since there will be no time to calculate them once the write operation has been initiated. This software uses a 16 bit cyclic redundancy check (CRC) word calculated from the data bytes by the binary polynomial:

$$x^{16} + x^{12} + x^5 + 1.$$

Because they are precalculated, the error detection bytes may be put in any position before, within, or after the data. I chose to place them directly after the data. The assembled block of data to be transferred (consisting of the preamble, sync byte, data and CRC bytes) is now in the proper format for transmission to the disk, but the disk drive itself is not yet properly set up for the transfer. Because the interface may control multiple drives, the first action of the software is to select the proper drive and to assure that it is ready. "Ready" means that a diskette is loaded and revolving, and the power supplies are working. Next, the current track location of the data transfer head is determined and compared to the desired track. If the desired and current tracks differ, the software must step the head in or out at the proper rate (10 ms per track) until the correct track is reached, and then delay for the proper head settling time (10 ms) before continuing.

If the head is not currently loaded, the software must load the head and allow time for the loading to be accomplished (30 ms).

All that remains before initiation of the data transfer is to find the starting point of the proper sector. Because the timing is fairly critical between finding the sector and initiating the write operation, all of the parameters for the write data loop such as pointers, counters and output commands must be set up ahead of time in the 8080 registers. The software now searches for the index pulse from the index hole of the selected disk drive and, having found it, begins to count sector pulses until the desired sector is found. When the leading edge of this sector pulse is sensed, the software transfers immediately to the write routines, using the values previously stored in the registers to speed the initiation of the write operation. This generates the proper timing relationship between the sector pulse and the start of the recorded data.

Write Data Transfer

If N bytes of information (including the cyclic redundancy check) are to be recorded on the disk, the software will actually send out N+33 bytes of data to the interface. Figure 1 shows that the first 16 bytes (a preamble of zero bytes) are recorded to allow the interface to correctly distinguish between data pulses and clock pulses when this data block is read. The seventeenth byte recorded is the sync byte, in this case a binary value of 10000001. This sync byte is used by the Universal Synchronous Receiver Transmitter (USRT) to find the boundaries between bytes during a read operation. The next N bytes are the block of data to be recorded, and finally there are 16 bytes as a postscript to assure that no data is destroyed when the disk drive write gate is turned off. Because the only use of this data is to maintain clock synchronization and protect the preceding data, the contents of the last 16 bytes are not critical, and may overlap data used for some other purpose.

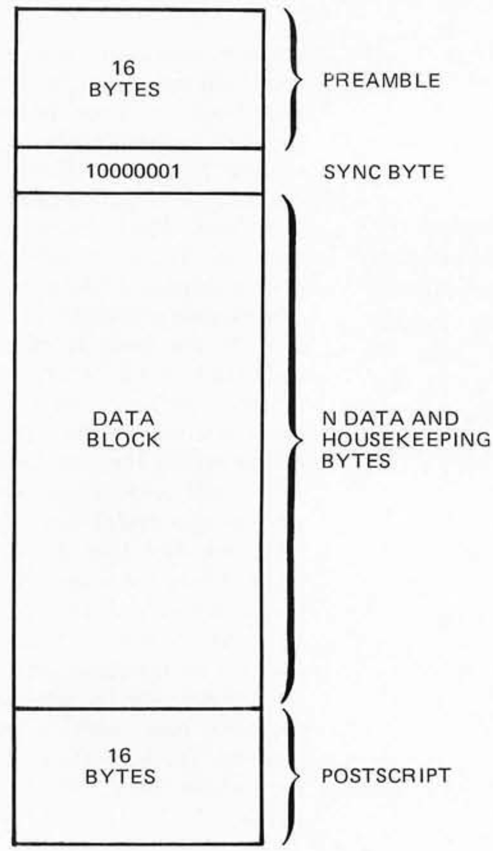


Figure 1: Floppy disk data transfer block format. The simplicity of this "hard sectored" floppy disk system requires that a specific format be used for the data. The first 16 bits act as a preamble to enable the interface to distinguish between data pulses and clock pulses. Next comes the sync byte used by the Universal Synchronous Receiver Transmitter (USRT) to find the boundaries between bytes. This is followed by the actual data and, finally, a 16 byte safety region to assure that no data is destroyed when the disk drive write head is turned off. See also table 2.

Write Error Detection

After all N+33 bytes have been sent to the disk through the interface, the write gate of the disk drive is turned off and error detection may now be performed. Error detection consists of performing a single read operation of the block of data just written. The block of data read in is compared byte by byte to the block of data written. If all N bytes are not the same for both blocks, an error has occurred. This process could be called a "verify" operation.

Write Error Correction

If error correction is necessary on a write operation that fails to verify, it is accomplished by rewriting the same block of data to the same disk, track and sector. After this rewrite attempt, a verify of the rewritten data is performed and compared to the correct data to determine the success or failure of the error correction operation. If

As block sizes increase there are fewer preambles and postscripts on any given track, thus maximizing the usable data bytes per track.

In the very simplest system each block of data would have some form of error detection ranging from a single byte of checksum to a 16 bit cyclic redundancy code or even a complex Hamming code.

the write operation fails four (or some other small integer) times in a row, all hope is abandoned, an error message is printed on the user's console device, and the write routine terminates. Manufacturers' recommendations for action taken at this point are as follows: Move to an unused sector and track on the disk in question, and retry the write operation. If the error persists, the disk drive has write circuitry problems; notify the user. If the error is eliminated, then the track and sector where the original error occurred probably has some damage to its oxide coating. In this case, relink the data file to reflect the new location of the data (the track and sector where the second attempt was made), and then record in some table the fact that the original track and sector where the write operation failed is an unusable area. But what do you do when the area that failed to write properly is the location of the table where the failed areas are stored? This is why operating systems designers have such a high incidence of insanity. The software in this article ignores the whole issue, the traditional ostrich solution.

Read Set Up

No data preparation need be done on a read operation because the only routines required are disk drive select, head load, track seek and sector seek. These are the same routines used by the write operation as described earlier.

Read Data Transfer

The USRT is reset to search for an occurrence of the sync byte within the incoming serial data, and the program is set to wait for the USRT to signal that the sync byte is found. When the sync byte is found, data is brought in from the interface and stored in memory. For a block recorded with N bytes of information as in the write operation, a total of N+1 bytes are read in. The first byte brought in by the software is the sync byte, accounting for the extra byte transferred.

Table 1: Characteristics of different data block sizes. Smaller data blocks have the advantage of not tying up large blocks of memory (a premium commodity in many small systems). Large data blocks, by comparison, speed up data transfer, require fewer blocks per track, and maximize the number of usable data bytes per track.

Data Blocks Per Track	Usable Bytes Per Block	Data Bytes Per Block	Housekeeping Bytes Per Block
1	5127	4096	1031
2	2549	2048	501
4	1260	1024	236
8	615	512	103
16	293	256	37
32	132	128	4

All Formats Store 315,392 Data Bytes Per Disk

Read Error Detection

The routine that calculated the 16 bit cyclic redundancy check (CRC) for the write operation is now used to calculate the CRC of the data block just read. The calculated cyclic redundancy check must match the CRC read in if the data is correct. If these values do not agree, an error has occurred during the read operation.

Read Error Correction

If a single read operation is unsuccessful, two more attempts are made to read the data. If the computed and read-in cyclic redundancy check values still disagree, the disk head is stepped in one track and then out one track, and up to three more read operations are tried. Continued lack of success causes the head to be stepped out and then stepped in one track, and three more attempts to read are made. If the error persists after all of this, then this software concludes that the data is unrecoverable. The jogging of the head one track in or out is recommended by the manufacturer to release any dust particles which may have lodged between the disk and head, and which therefore may be causing the read errors.

Block Size

The discussion of the software to this point has referred to the transfer of a "block" of data, but no definition has been made of the size of this block. The disk is divided physically into 32 different sectors by the presence of 32 sector holes. The combination of the rotational speed of the disk (360 rpm), the data rate (250,000 bps), variation in the timing of the sector hole detection ($\pm$ 500 μs), and the necessity for the 16 byte preamble, sync byte and 16 byte postscript on each record determines that if each track holds 32 blocks of data, then each block has 132 usable data bytes. There is no electronic or philosophical reason that the disk must be recorded in a format of 32 data blocks per track, or even that a 32 block track must consist of full size 132 byte blocks. The proper operating system software could quite easily make use of a data format with only 16 bytes per block and 11 blocks per track. However, this would make the capacity of each disk less than 30 K bytes, a tenfold reduction in storage capacity. The software presented here can be easily modified to allow other block sizes and formats to be used, and the reasons for choosing different block sizes will be presented before attempting to justify the block size used.

Smaller data blocks have three main advantages. First, because of the nature of a disk drive, data must be transferred one block at a time, never as a fraction of a block. If the data is not ready all at once (as is usually the case), then some area of computer memory must be dedicated to the storage of this data until a full block can be acquired. The size of this buffer area is the same as the size of the block, and a smaller data block consumes less memory than a large data block. Even with the dropping prices of memory, the average person cannot afford to dedicate large blocks of a computer's storage solely to disk data buffering, especially in a sophisticated operating system that may work on many disk data files simultaneously and require a data buffer for each one. Secondly, when recording data files onto a disk device, the data file rarely contains the right number of bytes to be stored in exactly an integral number of blocks. An average of one-half of a data block is wasted on each different file recorded on the disk. If one works with a data base that consists of a large number of short data files, this lost capacity of this disk can become significant. Smaller block size minimizes this loss of storage capacity. Last and definitely not least, the growing awareness of the need for standardization of data storage among the users is a motivation to seek out de facto (read IBM) standards to work with. Operating with somebody else's block and physical recording parameters makes it potentially possible to exchange data between different computer systems via diskettes. The most widely embraced standard for floppy disk data operations is the IBM soft sectored standard. Unfortunately, this standard is incompatible with the inexpensive interface I described in my previous article, and does not (at the time of this writing) lend itself to as cheap and simple an interface. However, the IBM standard uses 128 byte data blocks, and large amounts of software exist for it. Software for the hard sectored version of this format currently uses data blocks of 128 byte size, although the new double density disk drives are promoting software with 256 byte data blocks. Most existing software, then, is written with 128 byte blocks of data, 32 blocks per track on a hard sectored disk or 26 blocks per track on a soft sectored disk. This is an argument in favor of short data blocks, based on conventions of existing users of floppies.

Larger block sizes also have three main advantages. First, because the software presented here can only transfer one block of data per revolution, large data blocks greatly speed up data transfer. Dedicating a track to a single data block, a 5 K byte block of data can be transferred in 1/6 second. Second, large block sizes mean fewer blocks per disk. This reduces the data required in the addressing of disk data blocks, and can reduce the complexity of directories and block occupancy bit maps. Table 1 shows the third advantage of large data block sizes. As block sizes increase there is less overhead on any given track, and so the maximum number of usable data bytes per track (and per disk) increases. While the overall amount of data storage increases by only a small percentage, the increase is important to certain data structures, as will be shown.

In block structured random access data bases, such as disk storage, data is usually stored in blocks of 2^n bytes. An integral power of two is a handy size for organization of data in a binary computer and makes data manipulation easier than with some other sizes. With each block of useful data, there is some other data which is transparent to the user and of no real concern to him/her, but which is associated with file management and the operating system of the computer. This housekeeping or "overhead" data contains information about the status of the data within the block. In the very simplest system, for instance, each block of data would have some form of error detection, ranging from a single byte of checksum to a 16 bit cyclic redundancy code or even a complex error correcting Hamming code. Most simple systems also record the track number and sector number of the block with

No other peripheral device requiring direct memory access (DMA) operations should be active during the read or write operations of this software. In general, no interrupts or hold operations lasting longer than two microseconds can occur simultaneously with disk writing.

Table 2: This is an expansion of figure 1 showing how the data block is laid out in memory prior to an output disk transfer or following a read operation from the disk. The "housekeeping" bytes mentioned in the text are used for error detection and error correction as well as for labeling files.

Byte Number (Decimal)	Value	Function
−17 to −2	0	Clock synchronization
−1	hexadecimal 81	Sync byte for data synchronization
0 to 8	ASCII	9 letter file name
9	0 to 7	Drive number
10	3 to 76	Track number
11	0 to 30	Sector number (must be even)
12	3 to 76	Track number of preceeding block
13	0 to 30	Sector number of preceeding block
14	3 to 76	Track number of following block
15	0 to 30	Sector number of following block
16 to 19	—	Future use
20	0 to 255	Byte count of incomplete data blocks
21 to 276	DATA	User data (256 bytes)
277	—	Future use
278 to 279	CRC	16 bit CRC (cyclic redundancy check)
280 to 295	0	Trailer bytes

Bytes 0 to 279 are shown as they exist in memory.
Bytes −17 to 295 are shown as they exist on the disk.

NOTE: The software in this article uses only bytes 9 to 11 and 278, 279. (All bytes numbered 0 to 279 are transferred to disk and back.) See note at end of article concerning a complete floppy disk operating system which uses all the bytes.

Listing 1: Floppy disk drive input and output routines. As it stands, this listing is one step short of being a complete operating system, but could be used as a starting point for such an operating system. This program will run only on an 8080 with memory having a cycle time of 500 ns or faster; it will not run correctly if any interrupts or direct memory access (DMA) routines lasting longer than 2 µs occur during program execution. (See part 1, page 42, February 1977 BYTE, "Software Timing.")

```
               ; THE FOLLOWING IS A GENERAL PURPOSE ERROR TYPEOUT ROUTINE
               ; THE VALUE OF CO REFERS TO A USERS ROUTINE WHICH TYPES OUT
               ; THE ASCII VALUE IN THE C REGISTER ONTO THE USERS CONSOLE OUTPUT.
               ; ONLY THE A REGISTER AND FLAGS MAY BE DESTROYED BY CO.

0000           CO      EQU     0

0010                   ORG     10H      ;THIS ROUTINE MAY BE ORG'D ANYWHERE IN MEMORY

               ;THIS ROUTINE TYPES OUT AN ERROR MESSAGE TO THE USER.
               ;IT IS CALLED INTERNALLY AS:
               ;       CALL    ERTYP
               ;       DB      X
               ; THE RESULT IS THE TYPING OF:
               ;ERROR X

0010 E3        ERTYP:  XTHL
0011 F5                PUSH    PSW
0012 C5                PUSH    B
0013 D5                PUSH    D
0014 7E                MOV     A,M
0015 23                INX     H
0016 E5                PUSH    H
0017 F5                PUSH    PSW
0018 212800            LXI     H,ERMES
001B CD5500            CALL    TXTYP
001E F1                POP     PSW
001F CD2F00            CALL    BYTYP
0022 E1                POP     H
0023 D1                POP     D
0024 C1                POP     B
0025 F1                POP     PSW
0026 E3                XTHL
0027 C9                RET

0028 0D0A4552 ERMES:   DB      13,10,'ERRO','R'+128
002C 524FD2

002F F5        BYTYP:  PUSH    PSW
0030 0E20              MVI     C,' '
0032 CD0000            CALL    CO
0035 F1                POP     PSW
0036 F5        NMTYP:  PUSH    PSW
0037 0F                RRC
0038 0F                RRC
0039 0F                RRC
003A 0F                RRC
003B CD4300            CALL    HEXCHR
003E F1                POP     PSW
003F CD4300            CALL    HEXCHR
0042 C9                RET

0043 F5        HEXCHR: PUSH    PSW
0044 E60F              ANI     0FH
0046 C630              ADI     '0'
0048 FE3A              CPI     '9'+1
004A DA4F00            JC      HEX1
004D C607              ADI     'A'-'9'-1
004F 4F        HEX1:   MOV     C,A
0050 CD0000            CALL    CO
0053 F1                POP     PSW
0054 C9                RET

0055 7E        TXTYP:  MOV     A,M
0056 B7                ORA     A
0057 C8                RZ
0058 F5                PUSH    PSW
0059 E67F              ANI     7FH
005B 4F                MOV     C,A
005C CD0000            CALL    CO
005F F1                POP     PSW
0060 F8                RM
0061 23                INX     H
0062 C35500            JMP     TXTYP

0001           ERWRT   EQU     1        ;IF THE DISK FAILS TO WRITE,  TYPE "ERROR 1"
0002           ERRED   EQU     2        ;IF THE DISK FAILS TO READ,   TYPE "ERROR 2"
0003           ERDKNM  EQU     3        ;IF NO SUCH DISK EXISTS,      TYPE "ERROR 3"

               ; THESE ARE THE FLOPPY DRIVE I-O ROUTINES
               ;       19-AUG-76

00F0           INPORT  EQU     0F0H
00F0           OTPORT  EQU     0F0H

00F0           FDRED   EQU     INPORT          ;DATA FROM THE FLOPPY
00F1           FDSTAT  EQU     INPORT+1        ;STATUS BITS OF THE FLOPPY
00F2           SRTSWE  EQU     INPORT+2        ;READ THE SRT STATUS WORD
00F3           SRTRR   EQU     INPORT+3        ; INPUTTING THIS RESETS THE USRT TO LOOK
00F4           FDINWT  EQU     INPORT+4        ;KLUGE TO SYNCHRONIZE TO 32 MICRSECS

00F0           FDWRIT  EQU     OTPORT          ;DATA TO BE WRITTEN TO THE FLOPPY
00F1           SRTTFS  EQU     OTPORT+1        ;PORT OF THE DEFAULT CHARACTER
00F2           SRTRSS  EQU     OTPORT+2
00F3           FDOUT   EQU     OTPORT+3        ;SEND SIGNALS TO THE FLOPPY
00F4           FDOTWT  EQU     OTPORT+4        ;SYNC TO 32 MICROSECS
00F5           LOADHD  EQU     OTPORT+5        ;THIS COMMANDS LOADS THE HEAD FOR 3 SECONDS
00F6           HDUNLD  EQU     OTPORT+6        ;THIS FORCES AN IMMEDIATE HEAD UNLOAD

               ; THESE ARE SIGNALS TO FDOUT (DIRECT FLOPPY CONTROLS)
0001           WCSEL   EQU     1        ;1=TRACK 0-43, 0=TRACK 44-76
0002           FUREST  EQU     2        ;USE TO RESET A FILE UNSAFE CONDITION
0004           DIRIN   EQU     4        ;LOGICAL OR FOR IN TRACK
FFFB           DIROT   EQU     NOT DIRIN        ;AND FOR OUT TRACK
0008           WRTGAT  EQU     10Q      ;LOGICAL OR TO START A WRITE
0010           STEPP   EQU     20Q      ;LOGICAL OR TO START A STEP PULSE
FFEF           STEPM   EQU     NOT STEPP        ;LOGICAL AND TO STOP THE PULSE

               ; THESE ARE THE DIRECT STATUS LINES OF THE FLOPPY (IN FDSTAT)
0001           TRKZRO  EQU     1
0002           UNSAFE  EQU     2
0004           FDRDY   EQU     4
```

the rest of the data on that block. When some track and sector is read, the track and sector numbers read must agree with the desired track and sector numbers. If these bytes disagree, then a seek error has occurred. The error detection information and the track and sector numbers are simple housekeeping data that the user need never be concerned with during data transfer. A more complex data base management system may store extensive housekeeping information with each block of data. Information such as forward and reverse linkages (pointers to the succeeding and preceding blocks of data in a file), file name, number of valid bytes in a partially filled data block, date and time of the recording of this block of data, write or read protection, user identification, destroy date – all may be recorded as useful housekeeping data.

If a format of 32 blocks per track, 128 data bytes per block is used, table 1 shows that this leaves only four bytes of data available with each block to record housekeeping data. Going to 16 blocks of 256 data bytes per block, the available housekeeping area increases to 37 bytes per block. 512 data byte blocks gives a substantial 103 bytes of housekeeping on each of eight blocks per track. The routines presented here were written as a first low level step toward a full floppy disk operating sytem, and the amount of housekeeping required for the full system necessitated at least the 256 byte block size. The 512 byte block size was ruled out because it consumed buffer space much too quickly, and because a single byte would no longer be sufficient as a buffer data counter.

A DOS Block Format

Table 2 shows the layout of the data block for my disk operating system (DOS) as it exists in memory and as it is written to and read from the disk. There are 280 bytes in the block, 256 of which are usable data bytes, and 24 of which are reserved for housekeeping. Of the housekeeping bytes, five are not currently defined, and all but bytes 9, 10, 11, 278 and 279 are available to the user in the routines listed in this article. As of this writing, a full floppy disk operating system for my peripheral interface to the 8080 has been completed (see note at end of this article). The format described here is that used by this operating system. In this format, the first nine bytes are designated as the file name, ordered a la DEC as six ASCII characters of file name and three characters of extension. Having the file name associated with each block of a file is a great aid in data recovery from crashed systems (a not unheard of occurrence) and in troubleshooting modifications to the operating system (a frequently heard of occurrence). Byte 9 is the number of the device (disk 0 to disk 7) where this file was originally recorded. This number does not have to match the number of the device that the file is subsequently read from. Bytes 10 and 11 are the numbers of the track and sector where this data block is stored. Valid values are any integer from 0 to 76 for the track, and only even integers from 0 to 30 for the sector. (With 256 byte blocks, we have to use every other sector.) Bytes 12 and 13 indicate the track and sector of the block of data preceding this one in a linked data file. If the current block is the first block, and no previous data blocks exist, then bytes 12 and 13 are set to zero. Since track zero, sector zero is reserved for the bootstrap copy of the operating system, no linked file can have this block address as a legal forward or reverse linkage. Bytes 14 and 15 give the track and sector of the data block following this block in the linked file. If this is the last block, values of 0 and 0 are again used to indicate an invalid linkage (end of file).

Because no operating system is ever finished and better ideas are generated as a continuous function of time, bytes 16 to 19 are reserved for whatever functions may become desirable in the future. Byte 20 is used when the last block in a recorded data file is not completely filled, indicating the actual number of valid data bytes stored in the data area. Bytes 21 to 276 inclusive contain the 256 bytes of usable data. The data in these bytes is usually the only data of final concern to a program retrieving data from a file. Bytes 278 and 279 are the cyclic redundancy check bytes, least significant byte first. This 16 bit value is calculated as detailed earlier, from the values of bytes 0 to 276 inclusive.

Now, the Software

Listing 1 shows the software which controls the disk interface and performs the

Listing 1, continued:

```
0008            SCTR    EQU     8
0010            INDX    EQU     10H
0020            HEADLD  EQU     20H

                ; THESE ARE VARIOUS PARAMETERS
0081            SYNCB   EQU     81H     ;THIS IS THE SYNC BYTE FOR START OF TRACK
0100            LNBUF   EQU     256     ;LENGTH OF THE BUFFER
0016            XTRA    EQU     22      ;EXTRA BYTES FOR HOUSEKEEPING
0117            LN2SNC  EQU     LNBUF+XTRA+1    ;DISTANCE OF SYNC TO CHECKSUM
0008            DSKLIM  EQU     8       ;UP TO 8 DISK DRIVES ON THIS CONTROLLER

E000                    ORG     0E000H  ;THIS IS DISCONTIGUOUS RAM

E000 F3         GDWRT:  DI              ;DISABLE THE INTERRUPT TO MAINTAIN CRITICAL TIMING
E001 0602               MVI     B,2     ;TRY THIS 3 TIMES BEFORE GIVING UP
E003 C5         GDWR1:  PUSH    B
E004 CD15E0             CALL    WRT256  ;FIRST, WRITE IT
E007 CD6FE0             CALL    CHKWRT  ;THEN CHECK FOR PROPER WRITE
E00A C1                 POP     B       ;RESTORE STACK, DON'T CHANGE FLAGS
E00B C8                 RZ              ;SUCCESSFUL WRITE, RETRUN NOW
E00C 05                 DCR     B       ;A BAD WRITE, TRY AGAIN
E00D F203E0             JP      GDWR1   ;IF WE STILL HAVE TIME, TRY AGAIN
E010 CD1000             CALL    ERTYP   ;ELSE, SIGNAL THE ERROR AND QUIT
E013 01                 DB      ERWRT
E014 C9                 RET

                ;THIS ROUTINE ASSUMES THAT THE TRACK AND SECTOR TO BE WRITTEN ARE IN TRKWRT
                ; AND THE DATA IS ALL IN WRDAT
E015 3A65E2     WRT256: LDA     WRDEV   ;THIS MUST CONTAIN THE DISK NUMBER!!!
E018 CD2BE1             CALL    DSKNUM  ;PROCESS ANY UNIT NUMBER CHANGE
E01B 214BE2             LXI     H,WRTBUF        ;CLEAR THE CLOCK SYNC BYTES
E01E AF                 XRA     A       ;BY FILLING WITH ZEROES
E01F 0610               MVI     B,10H   ;16 OF THEM
E021 77         CLRLOP: MOV     M,A
E022 23                 INX     H
E023 05                 DCR     B
E024 C221E0             JNZ     CLRLOP
E027 3681               MVI     M,SYNCB ;STORE THE SYNC BYTE
E029 2A66E2             LHLD    TRKWRT
E02C 2248E2             SHLD    TRKWNT  ;SET UP THE TRACK AND SECTOR POINTS
E02F 215BE2             LXI     H,WRSNCB        ;START OF THE CHECKSUM AREA
E032 CDF3E0             CALL    CHK436
E035 EB                 XCHG
E036 2272E3             SHLD    WRCHKS  ;STORE THE CALCULATED CHECKSUM
E039 CDD8E1             CALL    TRKGET  ; GET THE DESIRED TRACK
E03C CD09E2             CALL    HEDLOD  ;LOAD THE HEAD
E03F 113601             LXI     D,LNBUF+XTRA+32
E042 214BE2             LXI     H,WRTBUF        ;BUFFER AREA TO WRITE TO THE DISK
E045 AF                 XRA     A
E046 D3F1               OUT     SRTTFS  ;TRANSMIT A FILL CHAR OF ZERO
E048 3A46E2             LDA     FDBUF   ;THE OUT STATUS
E04B F60A               ORI     WRTGAT+FUREST   ;TURN ON         THE WRITE GATE
E04D 4F                 MOV     C,A     ;HOLD THE WRITE COMMAND IN C
E04E CD1AE2             CALL    SCTGET  ;GET THE PROPER SECTOR
E051 79                 MOV     A,C
E052 D3F3               OUT     FDOUT   ;INIT THE WRITE GATE QUICKLY
E054 E6FD               ANI     NOT FUREST      ;TURN OFF FILE UNSAFE RESET
E056 D3F3               OUT     FDOUT
E058 D3F4       WRTLOP: OUT     FDOTWT  ;KLUGE AND WAIT FOR READY FOR DATA
E05A 7E                 MOV     A,M
E05B D3F0               OUT     FDWRIT
E05D 23                 INX     H
E05E 1B                 DCX     D
E05F 7A                 MOV     A,D
E060 B3                 ORA     E
E061 C258E0             JNZ     WRTLOP
E064 3A46E2             LDA     FDBUF
E067.E6F7               ANI     NOT WRTGAT      ;TURN OFF THE WRITE GATE
E069 D3F3               OUT     FDOUT   ;NOW
E06B 3246E2             STA     FDBUF   ;THIS IS THE PRESENT STATUS
E06E C9                 RET

                ; THIS ROUTINE PERFORMS A REREAD OF THE SECTOR JUST WRITTEN.
                ;RETURN IS WITH Z=0 IF A READ FAIL OCCURRED, OR IF A READ DIDN'T MATCH
                ; THE IMMEDIATE PREVIOUS WRITE. Z=1 IS A SUUCCESSFUL WRITE
E06F CDA3E0     CHKWRT: CALL    TRY3RD  ;TRY 3 TIMES TO READ IT
E072 C0                 RNZ     ;RETURN WITH Z=0 ON A BAD ERROR
E073 2175E3             LXI     H,RDSNCB        ;SET UP A 256 BYTE COMPARE
E076 115BE2             LXI     D,WRSNCB
E079 0600               MVI     B,0     ;256 ISN'T ALL, BUT IT IS ENOUGH
E07B 1A         CHKW1:  LDAX    D       ;GET WHAT WAS WRITTEN
E07C BE                 CMP     M       ;COMPARE TO WHAT WAS READ
E07D C0                 RNZ             ;RETURN ON AN ERROR
E07E 13                 INX     D
E07F 23                 INX     H       ;UP THE POINTERS
E080 05                 DCR     B
E081 C27BE0             JNZ     CHKW1
E084 C9                 RET             ;SUCCESSFUL RETURN, Z=1

                ;THIS ROUTINE TRIES 3 TIMES TO READ, THEN JOGS IN, AND OUT AND AGAIN
E085 F3         RED256: DI              ;DISABLE THE INTERRUPT FOR CRITICAL TIMING
E086 CDA3E0             CALL    TRY3RD  ;TRY 3 TIMES TO READ THE BLOCK
E089 C8                 RZ      ;IF SUCCESSFUL, THEN RETURN
E08A CDA3E1             CALL    TRAKIN  ;JOG IN
E08D CDB2E1             CALL    TRAKOT  ;SCRAPE OFF FLIES AND FROGS
E090 CDA3E0             CALL    TRY3RD  ;THREE MORE
E093 C8                 RZ
E094 CDB2E1             CALL    TRAKOT  ;JIG OUT
E097 CDA3E1             CALL    TRAKIN  ;SCRAPE OFF DIGITS AND DOGS
E09A CDA3E0             CALL    TRY3RD  ;LAST CHANCE COWBOY
E09D C8                 RZ              ;IF FINALLY SUCCESSFUL, THEN RETURN
E09E CD1000             CALL    ERTYP   ;IF NOT, THEN TYPE OUT THE ERROR
E0A1 02                 DB      ERRED
E0A2 C9                 RET

                ; THIS ROUTINE TRIES TO READ THREE TIMES, RETURNS Z=1 FLAG      IF OK ON REA
E0A3 3E02       TRY3RD: MVI     A,2     ;ACTUALLY TRY 3 TIMES
E0A5 3245E2             STA     REDTRY  ;HOLD ONTO IT
E0A8 CDB4E0     TRGP:   CALL    REDONC  ;TRY IT JUST ONCE
E0AB C8                 RZ
```

Listing 1, continued:

```
E0AC 2145E2          LXI     H,REDTRY
E0AF 35              DCR     M
E0B0 F8              RM              ;IF TOO MANY, THEN RETURN Z=0
E0B1 C3A8E0          JMP     TRGP    ;TRY TRY AGAIN, AGAIN

              ;THIS ROUTINE TRIES ONLY ONCE (AND I DON'T BLAME IT ATALL)
E0B4 3A4AE2   REDONC: LDA    DSKWNT  ;THIS MUST CONTAIN THE DISK NUMBER!!
E0B7 CD2BE1          CALL    DSKNUM  ;PROCESS ANY UNIT CHANGE
E0BA CDD8E1          CALL    TRKGET  ;GET THE TRACK
E0BD CD09E2          CALL    HEDLOD  ;LOAD THE HEAD
E0C0 111A01          LXI     D,LNBUF+XTRA+4
E0C3 2175E3          LXI     H,RDSNCB        ;POINT TO THE BUFFER AREA
E0C6 3E81            MVI     A,SYNCB ;PROMPT THE USRT WITH WHAT TO EXPECT
E0C8 D3F2            OUT     SRTRSS  ;SHOVE IT
E0CA CD1AE2          CALL    SCTGET  ;GET THE PROPER SECTOR
E0CD DBF3            IN      SRTRR   ;RESET THE USRT TO LOOK FOR THE SYNCB
E0CF DBF4     REDLOP: IN     FDINWT  ;SPECIAL SYNC KLUGE TO HOLD ONTO THE BUD
E0D1 DBF0            IN      FDRED   ;UNTIL DATA IS AVAILABLE
E0D3 77              MOV     M,A
E0D4 23              INX     H
E0D5 1B              DCX     D
E0D6 7A              MOV     A,D
E0D7 B3              ORA     E
E0D8 C2CFE0          JNZ     REDLOP
E0DB 2175E3          LXI     H,RDSNCB        ;POINT TO THE BUFFER AREA
E0DE CDF3E0          CALL    CHK436
E0E1 3A3AE2          LDA     LSTDSK  ;GET THE NUMBER OF THIS DISK,
E0E4 327FE3          STA     RDDEV   ;AND STORE IT IN THE READ AREA FOR FUTURE USE
E0E7 2A8CE4          LHLD    RDCHKS  ;HL HAS THE CHECKSUM
E0EA C3EDE0          JMP     COMPAR

              ;THIS ROUTINE COMPARES HL TO DE
E0ED 7C       COMPAR: MOV    A,H
E0EE 92              SUB     D
E0EF C0              RNZ
E0F0 7D              MOV     A,L
E0F1 93              SUB     E
E0F2 C9              RET

              ;THIS ROUTINE DOES A DOUBLE PRECISE CHECKSUM
E0F3 011601   CHK436: LXI    B,LNBUF+XTRA
E0F6 110000   CHKSUM: LXI    D,0
E0F9 7E       CHKLOP: MOV    A,M     ;GET THE BYTE TO SUM
E0FA E5              PUSH    H
E0FB C5              PUSH    B       ;SAVE ALL THE REGS.
E0FC AB              XRA     E
E0FD 47              MOV     B,A
E0FE 0F              RRC
E0FF 0F              RRC
E100 0F              RRC
E101 0F              RRC
E102 4F              MOV     C,A
E103 A8              XRA     B
E104 E6F0            ANI     0F0H
E106 AA              XRA     D
E107 6F              MOV     L,A
E108 79              MOV     A,C
E109 07              RLC
E10A E61F            ANI     1FH
E10C AD              XRA     L
E10D 6F              MOV     L,A
E10E 78              MOV     A,B
E10F 07              RLC
E110 E601            ANI     1
E112 AA              XRA     D
E113 AD              XRA     L
E114 57              MOV     D,A
E115 79              MOV     A,C
E116 E60F            ANI     0FH
E118 A8              XRA     B
E119 5F              MOV     E,A
E11A 79              MOV     A,C
E11B A8              XRA     B
E11C 07              RLC
E11D E6E0            ANI     0E0H
E11F AB              XRA     E
E120 5F              MOV     E,A
E121 C1              POP     B
E122 E1              POP     H
E123 23              INX     H
E124 0B              DCX     B
E125 78              MOV     A,B
E126 B1              ORA     C
E127 C2F9E0          JNZ     CHKLOP  ;NEXT BYTE IF NOT YET DONE
E12A C9              RET

              ;THIS ROUTINE IS ENTERED WITH A CONTAINING THE NUMBER OF THE
              ; DISK DRIVE UNIT ON WHICH THE DESIRED OPERATIONS ARE TO BE PERFORMED.
              ; (0-DSKLIM-1).  IF THE LAST DRIVE USED WAS THIS DRIVE, THEN
              ; RETURN IS IMMEDIATE, AND NO ACTION IS TAKEN.  IF A NEW DRIVE
              ; IS CALLED, THEN THE HEAD OF THE PRESENT DRIVE IS FORCED TO UNLOAD
              ; AND THE CURRENT TRACK NUMBER ASSOCIATED WITH IT IS STORED.
              ; THE NEW DISK'S LAST TRACK IS REMEMBERED, AND IF VALID, CONTROL IS
              ; RETURNED. ELSE, THE NEW DISK IS INITED, AND THE TRACK 0 IS FOUND.

E12B FE08     DSKNUM: CPI    DSKLIM  ;0-7 IS CONDITIONALLY ACCEPTABLE
E12D F285E1          JP      DSKR1   ;NO SUCH DISK EXISTS!!
E130 213AE2          LXI     H,LSTDSK        ;POINTER TO THE LAST DISK
E133 BE              CMP     M       ;NEW=OLD?
E134 C8              RZ              ;IF SO, THEN RETURN NO ACTION
E135 F5              PUSH    PSW     ;ELSE, SAVE THE NEEDED DATA
E136 E5              PUSH    H
E137 3E08            MVI     A,DSKLIM        ;WAS THE OLD ONE LEGAL?
E139 BE              CMP     M
E13A FA46E1          JM      DSKN1   ;IF NOT, DON'T STORE THE OLD TRACK
E13D 4E              MOV     C,M     ;C=LAST NUMBER
E13E 0600            MVI     B,0
E140 09              DAD     B
E141 23              INX     H       ;HL=LSTDSK+OLD DISK+1
```

operations described above. There are three main entry points to this software: DKINT, GDWRT and RED256. When called, these routines initialize a disk drive, write a data block, and read a data block, respectively.

DKINT

DKINT (for DisK INiTialize) is a subroutine called with the accumulator containing the number (0 thru 7) of the disk drive to be initialized. DKINT causes the selected disk drive to unload its data transfer head, step the head to track zero, and reset the track counter associated with that drive to zero. DKINT destroys the contents of all of the 8080 registers, so appropriate precautions should be taken when using it. This routine must be used after powering up the processor, and should be used both before disks are removed from the disk drives, and before they are referenced by programs after insertion.

GDWRT

GDWRT (for GooD WRiTe) causes a data block to be written to a specified track and sector on a specified disk. When the GDWRT subroutine is called, all of the data relevant to the transfer is assumed to already reside in the 280 byte buffer area from WRFLNM to WRFLNM+279. The number of the disk to be written on (0 thru 7) must be in location WRDEV (which is WRFLNM+9), the track number (0 thru 76) must be in TRKWRT (which is WRFLNM+10), and the sector number (an even number from 0 to 30) must be in TRKWRT+1 (WRFLNM+11). The data to be stored on the disk may reside anywhere in bytes 0 to 8 and 12 to 277 of the buffer area, but in the context of my operating system, the 256 bytes from WRFLNM+21 to WRFLNM+276 are specifically reserved for data. GDWRT performs the steps previously described for the write data block operation: generating the preamble, calculating the cyclic redundancy check, selecting the disk and positioning the head, writing the block, verifying the write operation and attempting to rewrite if necessary. Control will return to the program which called GDWRT after a delay of from 0.2 to 2 seconds, depending on the distance that the head had to travel and the occurrence of any write errors. The user need not worry about the timing of the operation; however, no interrupts or hold operations longer than 2 μs may occur during GDWRT. Specifically, no memory slower than 500 ns response time may be used, and no device using direct memory access (DMA) operations may occur during GDWRT. If such a delay happens at the wrong time, the data will not be properly recorded, and following data blocks on the same track may even be overwritten and destroyed. Caution must also be observed when using both GDWRT and RED256 from the same program, since the write check operation of the GDWRT routine reads data into the buffer used by the RED256 routine, thereby destroying data previously present in the buffer. GDWRT performs the error detection and correction steps outlined above the returns with the zero flag true whenever the write operation was successful. If some write error occurs, the zero flag will be false (a JNZ instruction will jump). The contents of the 8080 registers are destroyed by GDWRT in either case, and only the Z flag returns with valid information.

RED256

RED256 (for REaD a 256 byte block) reads a data block in from a specified disk, track and sector. When RED256 is called, it initiates the read operation, and assumes that the disk, track and sector of the data block to be read are stored in the memory locations DSKWNT, TRKWNT and SECWNT, respectively, with the same restrictions on the disk track and sector values as in the GDWRT routine. After being called, RED256 will return to the calling program after a delay of from 0.1 to 3 seconds. Upon returning, the zero flag is set (true) to indicate that the data block read in had the proper cyclic redundancy check bytes, and that the read operation was successful. The zero flag is cleared (false) to indicate that the read failed to produce the proper check value even after the multiple reads and head jogs described in the read error correction section. After a successful read, the block of data read will be present in the 280 byte buffer area from RDFLNM to RDFLNM+279, in the format shown in table 2. Because the buffer area is also used to check for errors during a write operation, as noted previously, the data read from the disk by a call to RED256 should be copied into another buffer area if the data is to be kept beyond the time of the next data block read or data block write. None of the 8080 registers are sacred during a call to RED256, and the programmer must take this into account.

The three entry points described, DKINT, GDWRT and RED256, are sufficient for all operations involved in any data base management or operating system which uses the data block format of table 2. However, the personal computing enthusiast (by definition never satisfied) may wish to use different data formats or different disk drives.

Listing 1, continued:

```
E142 3A47E2           LDA     TRAKNO
E145 77               MOV     M,A      ;STORE THE OLD CURRENT TRACK NUMBER
E146 D3F6   DSKN1:    OUT     HDUNLD   ;FORCE THE HEAD TO UNLOAD
E148 E1               POP     H
E149 F1               POP     PSW      ;RETRIEVE THE OLD DATA
E14A 77               MOV     M,A      ;STORE THE NEW NUMBER AS THE OLD ONE
E14B 4F               MOV     C,A      ;BC=NEXT DISK
E14C 0F               RRC
E14D 0F               RRC
E14E 0F               RRC              ;A=NEWTRACK*32
E14F 3246E2           STA     FDBUF    ;CONSIDER THIS THE NEXT DRIVE COMMAND
E152 D3F3             OUT     FDOUT    ;SET UP AHEAD OF TIME TO DEGLITCH
E154 DBF1             IN      FDSTAT   ;CHECK THE STATUS OF THIS DISK
E156 E604             ANI     FDRDY    ;IS IT LOADED AND USEFUL?
E158 C285E1           JNZ     DSKR1    ;NO, TELL USER ABOUT IT!
E15B 09               DAD     B
E15C 23               INX     H
E15D 7E               MOV     A,M      ;THIS IS WHERE THE DISK WAS LEFT
E15E 3247E2           STA     TRAKNO   ;DECLARE IT THE CURRENT COUNT
E161 FE4D             CPI     77       ;0-76?
E163 F8               RM

E164 DBF1   GTRK0:    IN      FDSTAT
E166 E604   DSKN2:    ANI     FDRDY
E168 C284E1           JNZ     NODSK
E16B CDA3E1           CALL    TRAKIN
E16E CDA3E1           CALL    TRAKIN
E171 CDA3E1           CALL    TRAKIN
E174 CDA3E1           CALL    TRAKIN
E177 CDB2E1 TRZLP:    CALL    TRAKOT
E17A DBF1             IN      FDSTAT
E17C E601             ANI     TRKZRO
E17E C277E1           JNZ     TRZLP
E181 3247E2           STA     TRAKNO
E184 C9     NODSK:    RET

E185 CD1000 DSKR1:    CALL    ERTYP    ;DISK NUMBER >7
E188 03               DB      ERDKNM
E189 C9               RET

E18A 3EFF   INIT:     MVI     A,255
E18C 323AE2           STA     LSTDSK
E18F 3C               INR     A
E190 F5     INI1:     PUSH    PSW
E191 CD9CE1           CALL    DKINT
E194 F1               POP     PSW
E195 3C               INR     A
E196 FE08             CPI     DSKLIM
E198 C290E1           JNZ     INI1
E19B C9               RET

E19C CD2BE1 DKINT:    CALL    DSKNUM
E19F CD64E1           CALL    GTRK0
E1A2 C9               RET

            ;THESE ROUTINES MOVE THE TRANK
            ;THESE ROUTINES MOVE THE 'EAD IN (TOWARDS HUB) OR OUT (TOWARDS CIRCUMFER
            ;ONE TRACK FOR EACH CALL9        CALL LASTS 10 MILSECS
E1A3 2147E2 TRAKIN:   LXI     H,TRAKNO
E1A6 34               INR     M        ;COUNT UP ONE TRACK
E1A7 3A46E2           LDA     FDBUF
E1AA F604             ORI     DIRIN
E1AC 3246E2           STA     FDBUF
E1AF C3BEE1           JMP     TRAKMV
E1B2 2147E2 TRAKOT:   LXI     H,TRAKNO
E1B5 35               DCR     M        ;COUNT DOWN THE TRACK
E1B6 3A46E2           LDA     FDBUF
E1B9 E6FB             ANI     DIROT    ;CLEAR FOR AN OUT MOVE
E1BB 3246E2           STA     FDBUF    ;SAVE IT
E1BE F610   TRAKMV:   ORI     STEPP    ;SET THE STEP BIT
E1C0 D3F3             OUT     FDOUT    ;AND THE START BIT
E1C2 E6EF             ANI     STEPM    ;CLEAR STEP BIT
E1C4 D3F3             OUT     FDOUT
E1C6 CDCAE1           CALL    TENMIL   ;WAIT TEN NILSECS
E1C9 C9               RET
            ;THIS ROUTINE IS JUST A SOFTWARE 10MS WAIT
E1CA 01FF03 TENMIL:   LXI     B,1777Q
E1CD AF               XRA     A
E1CE 0B     TMLP:     DCX     B
E1CF B9               CMP     C
E1D0 C2CEE1           JNZ     TMLP
E1D3 B8               CMP     B
E1D4 C2CEE1           JNZ     TMLP
E1D7 C9               RET
            ;THIS ROUTINE GETS THE DESIRED TRACK (FOUND IN TRAWNT)
E1D8 3A48E2 TRKGET:   LDA     TRKWNT   ;GET THE DESIRED TRACK NUMBER
E1DB FE4D             CPI     77
E1DD F0               RP      ;ERROR
E1DE 2146E2           LXI     H,FDBUF
E1E1 7E               MOV     A,M
E1E2 E6FE             ANI     NOT WCSEL
E1E4 77               MOV     M,A
E1E5 3A48E2           LDA     TRKWNT
E1E8 DE2C             SBI     54Q
E1EA F2F1E1           JP      TRKG1
E1ED 7E               MOV     A,M
E1EE F601             ORI     WCSEL
E1F0 77               MOV     M,A
            TRKG1:
E1F1 3A47E2 TGETLP:   LDA     TRAKNO   ;GET THE PRESENT TRACK NUMBER
E1F4 47               MOV     B,A      ;B IS WHERE WE ARE AT
E1F5 3A48E2           LDA     TRKWNT   ;A IS WHERE WE WANT TO BE
E1F8 B8               CMP     B
E1F9 C8               RZ      ;TADA... WE ARE THERE
E1FA F203E2           JP      INGIN    ;A>B THEN GO IN
E1FD CDB2E1           CALL    TRAKOT   ;A.<B. THEN OUT
E200 C3F1E1           JMP     TGETLP
E203 CDA3E1 INGIN:    CALL    TRAKIN
E206 C3F1E1           JMP     TGETLP
            ;THIS ROUTINE LOADS THE HEAD AND WAITS 30 MS FOR SETTLING
```

Listing 1, continued:

```
E209 DBF1       HEDLOD: IN      FDSTAT
E20B E620               ANI     HEADLD
E20D D3F5               OUT     LOADHD
E20F C0                 RNZ
E210 CDCAE1             CALL    TENMIL
E213 CDCAE1             CALL    TENMIL
E216 CDCAE1             CALL    TENMIL
E219 C9                 RET
E21A DBF1       SCTGET: IN      FDSTAT
E21C E610               ANI     INDX
E21E C21AE2             JNZ     SCTGET  ;WAIT FOT THE INDEX PULSE
E221 3A49E2             LDA     SECWNT  ;GET THE DESIRED SECTOR
E224 E61E               ANI     30
E226 47                 MOV     B,A
E227 DBF1       SCTLP:  IN      FDSTAT
E229 E608               ANI     SCTR
E22B C227E2             JNZ     SCTLP
E22E 05                 DCR     B
E22F F8                 RM
E230 DBF1       SCTL2:  IN      FDSTAT
E232 E608               ANI     SCTR
E234 CA30E2             JZ      SCTL2   ;WAIT FOR END OF SECTOR PULEE
E237 C327E2             JMP     SCTLP

                ;ALL THE FOLLOWING MUST BE RAM

                ; ALL DATA FILES, LINKED OR CONTIGUOUS, SHALL HAVE THE FOLLOWING FORMAT:
                ;
                ;       BYTE 0-8        FILNAM.EXT, 9 BYTES ASCII FILE NAME ON EACH BLOCK
                ;       BYTE 9          DEVICE NUMBER, IGNORE
                ;       BYTE 10,11      CURRENT TRACK (3-76), SECTOR (0-30, EVEN) OF THIS BLOCK
                ;       BYTE 12,13      TRACK SECTOR OF PREVIOUS BLOCK (0,0 IF THIS IS FIRST)
                ;       BYTE 14,15      TRACK SECTOR OF FOLLOWING BLOCK (0,0 IF THIS IS LAST)
                ;       BYTE 16-19      FUTURE USE
                ;       BYTE 20         BYTE COUNT OF INCOMPLETE DATA BLOCKS
                ;       BYTE 21-276     256 DATA BYTES
                ;       BYTE 278-279    CYCLIC REDUNDANCY CHECK BYTES

E23A 00         LSTDSK: DB      0       ;NUMBER OF THE LAST DISK TO BE USED
E23B            DKTR0:  DS      DSKLIM  ;UP TO 8 DISKS WITH SEPERATE TRACKS TO KEEP
E243 0000       SCTNXT: DW      0       ;TEMPORARY POINTER TO NEXT SECTOR
E245 00         REDTRY: DB      0       ;COUNTER FOR 3 ATTEMPTES
E246 00         FDBUF:  DB      0       ;BUFFERE TO THE FD CONTROLS
E247 00         TRAKNO: DB      0       ;CURRENT TRACK
E248 00         TRKWNT: DB      0       ;DESIRED TRACK
E249 00         SECWNT: DB      0       ;DESIRED SECTOR NUMBER
E24A 00         DSKWNT: DB      0       ;DESIRED DISK NUMBER
E24B            WRTBUF: DS      16      ;128 BITS OF 0 SYNC'S THE CLOCK
E25B 81         WRSNCB: DB      SYNCB   ;SYNC BYTE GOES HERE
E25C 46494C4E WRFLNM:   DB      'FILNAM','EXT'  ;NAME OF FILE AS IT APPEARS IN THE DIRECTORY
E260 414D4558
E264 54
E265 00         WRDEV:  DB      0       ;ONE OF 16 DEVICES OF 16 TYPES
E266 0000       TRKWRT: DB      0,0     ;TRACK, SECTOR THAT THIS WILL BE WRITTEN TO
E268 0000       WRVLNK: DB      0,0     ;TRACK, SECTOR OF THE LAST BLOCK BEFORE THIS
E26A 0000       WFWLNK: DB      0,0     ;TRACK, SECTOR OF THE NEXT BLOCK AFTER THIS
E26C                    DS      4       ;RESERVE 4 BYTES FOR EXPANSION
E270 00         WRCNT:  DB      0       ;REALLY ONLY USED BY THE BYTOT ROUTINES
E271            WRDAT:  DS      256
E372            WRCHKS  EQU     WRSNCB+LN2SNC
E375                    ORG     WRCHKS+3
E375 81         RDSNCB: DB      SYNCB   ;HERE FALLS THE INPUT SYNCBYTE
E376 46494C4E RDFLNM:   DB      'FILNAM','EXT'  ;SAME AS DIRECTORY ENTRY
E37A 414D4558
E37E 54
E37F 00         RDDEV:  DB      0       ;INCONSEQUENTIAL
E380 0000       TRKRED: DB      0,0     ;THIS MUST AGREE WITH THE TRACK, SECTOR DESIRED
E382 0000       RRVLNK: DB      0,0     ;LAST TRACK SECTOR READ
E384 0000       RFWLNK: DB      0,0     ;NEXT TRACK SECTOR READ FOR THIS FILE
E386                    DS      4       ;EXPANSION
E38A 00         RDCNT:  DB      0
E38B            RDDAT:  DS      256
E48C            RDCHKS  EQU     RDSNCB+LN2SNC
0000                    END
BYTYP 002F      CHK43 E0F3      CHKLO E0F9      CHKSU E0F6
CHKW1 E07B      CHKWR E06F      CLRLO E021      CO    0000
COMPA E0ED      DIRIN 0004      DIROT FFFB      DKINT E19C
DKTR0 E23B      DSKLI 0008      DSKN1 E146      DSKN2 E166
DSKNU E12B      DSKR1 E185      DSKWN E24A      ERDKN 0003
ERMES 0028      ERRED 0002      ERTYP 0010      ERWRT 0001
FDBUF E246      FDINW 00F4      FDOTW 00F4      FDOUT 00F3
FDRDY 0004      FDRED 00F0      FDSTA 00F1      FDWRI 00F0
FUPES 0002      GDWR1 E003      GDWRT E000      GTRK0 E164
HDUNL 00F6      HEADL 0020      HEDLO E209      HEX1  004F
HEXCH 0043      INDX  0010      INGIN E203      INI1  E190
INIT  E18A      INPOR 00F0      LN2SN 0117      LNBUF 0100
LOADH 00F5      LSTDS E23A      NMTYP 0036      NODSK E184
OTPOR 00F0      RDCHK E48C      RDCNT E38A      RDDAT E38B
RDDEV E37F      RDFLN E376      RDSNC E375      RED25 E085
REDLO E0CF      REDON E0B4      REDTR E245      RFWLN E384
RRVLN E382      SCTGE E21A      SCTL2 E230      SCTLP E227
SCTNX E243      SCTR  0008      SECWN E249      SRTRR 00F3
SRTRS 00F2      SRTSW 00F2      SRTTF 00F1      STEPM FFEF
STEPP 0010      SYNCB 0081      TENMI E1CA      TGETL E1F1
TMLP  E1CE      TRAKI E1A3      TRAKM E1BE      TRAKN E247
TRAKO E1B2      TRGP  E0A8      TRKG1 E1F1      TRKGE E1D8
TRKRE E380      TRKWN E248      TRKWR E266      TRKZR 0001
TRY3R E0A3      TRZLP E177      TXTYP 0055      UNSAF 0002
WCSEL 0001      WFWLN E26A      WRCHK E372      WRCNT E270
WRDAT E271      WRDEV E265      WRFLN E25C      WRSNC E25B
WRT25 E015      WRTBU E24B      WRTGA 0008      WRTLO E058
WRVLN E268      XTRA  0016
```

The following subroutines, which are contained in this software, are useful for such purposes.

WRT256

This subroutine performs all of the write operations: generate preamble, sync byte, cyclic redundancy calculation, head load and seek, and data write. This is all that is done; no verify operation is performed, and the write is only performed once per call of WRT256.

REDONC

This subroutine performs all of the read operations: head load and seek, data read, cyclic redundancy check generation and comparison. Control is returned to the calling program with the Z flag set if the calculated cyclic redundancy check equals the value read from the disk. The read is performed only once per call of REDONC, regardless of whether the checking information indicates a proper read or not.

DSKNUM

In a multiple disk drive system, when changing from a read or write operation on one disk to a read or write operation on another drive, this routine must be called. The number of the next drive to transfer data to or from must be in the accumulator (a value of 0 to 7) when DSKNUM is called. If the number of the next drive to be used is different from that of the last drive, then DSKNUM will unload the head of the last drive, store the number of the track where the head of the last drive is positioned, enable the next drive, and recall the track where the head of the next drive is positioned. If the next and last drives are the same, no action is taken.

CHKSUM

This routine is entered with the HL register pair pointing to the first byte of some block of data whose cyclic redundancy check is to be calculated. The BC register pair contains some number from 1 to 65,536, indicating how many bytes are used in the calculation. The 16 bit check value is returned in the DE register pair. The program used to calculate this binary polynomial is adapted from the Intel Users Library (program 80-4I).

TRKGET

The desired track, from 0 to 76, is stored in the byte labeled TRKWNT. When TRKGET is called, the selected disk (selected by DSKNUM above) will move its data transfer head in or out until it is over the desired track. The desired track number is compared to 44, and the proper write current is selected on the WRITE CURRENT SELECT line of the interface. Upon returning from TRKGET, the head is at the proper track and the required head settling time has transpired. Calling TRKGET when the head is already at the proper track causes an immediate return; this will not significantly slow down the calling program.

TRAKIN

Calling this routine moves the data transfer head of the selected disk inward one track and increments the track counter. A 10 ms track movement delay occurs before control returns to the calling program.

TRAKOT

TRAKOT performs the inverse of TRAKIN by moving the head out one track, decrementing the track counter and delaying 10 ms before returning.

TENMIL

This is a software delay routine which delays 10 ms before returning. When using an 8080 or Z-80 system with a faster cycle time than 2 MHz, the delay loop counter should be changed. Because this software will not operate properly in memory with a cycle time longer than 500 ns, no values for slow memory need be given.

HEDLOD

This routine generates a head load pulse which causes the head of the selected disk drive to be loaded for at least 3 seconds following the call to HEDLOD. If the head was not loaded when HEDLOD was called, a delay of 30 ms occurs before control is returned to the calling program. If the head was loaded when HEDLOD was called, then the return is immediate.

SCTGET

SCTGET is called with the desired sector value in the memory location labeled SECWNT. The routine waits for an index pulse to occur on the selected disk drive, and then counts the sector pulses that follow. The first sector pulse following an index pulse is the start of sector 0, and the last pulse preceding an index pulse is the start of sector 31. Control is returned to the calling program within 15 to 25 μs of when the leading edge of the desired sector pulse is found. Only registers A and B are used by this routine, allowing all other registers to be used by the calling program.

Conclusion

The software described in this article allows the advanced computer experimenter, who has implemented the economy floppy disk interface, to make good use of his/her floppy disk drive(s) for data storage and retrieval. These routines have been operational for well over one year, and for the last year, GDWRT, RED256 and DKINT have formed the heart of my 4 K byte (including data buffer areas) 8080 floppy disk operating system. This operating system allows machine language programs to be stored and retrieved, MITS BASIC programs to be stored and loaded (through the CSAVE and CLOAD routines), editing of text files (this article and my previous one were written and edited on the disk system) and batch stream processing (by assigning the keyboard input to be taken from a batch stream file on the disk). An unlimited number of files may be open for input or output simultaneously, requiring only a 280 byte buffer area for each file. BASIC programs can create, read, write and modify data files in ASCII or binary. All files can be specified with a 9 letter file name and a disk number.

The rest is up to you. This sytem can give you the extra bit of programming power you've been looking for!■

Object Code in Machine Readable Form

The bar code representation of Dr Welles' floppy disk software is expected to be prepared in time for the July issue of BYTE, where it will be printed as a small feature supplementing the information in this month's article.

Artificial Intelligence,

Part 2: Implementation

Michael Wimble
6026 Underwood Av
Cedar Rapids IA 52404

As described last month in part 1, there are five types of mutations that can be performed with the simulated evolution technique. A separate subroutine will be used for each mutation type. Four of the subroutines rely heavily on a subroutine which generates a random number between limits. (For those systems not already possessing random number generators, a box accompanying this article gives an algorithm to produce pseudorandom numbers by the power residue method.)

Figures 1 through 9 are flowcharts of the basic modules which were extracted from a fairly sophisticated system of FORTRAN programs. These are intended to serve as a starting point for the reader in implementing his/her own program. If there is sufficient reader interest, I would be happy to program and publish program listings of implementations for one or more small system processors, in any popular computer language.

The rest of this article then is a description of the modules used to implement a 2 symbol gaming program using the artificial intelligence technique. If more detail is needed I suggest you look first into the book *Artificial Intelligence Through Simulated Evolution* by L J Fogel, A J Owens and M J Walsh (John Wiley and Sons, New York, 1966) or send questions and a self-addressed stamped envelope to me. Although there are many flowcharts and the technical jargon may seem complex at times, I wish to emphasize that the programming is simple and the technique can be implemented on any personal computing system with sufficient memory.

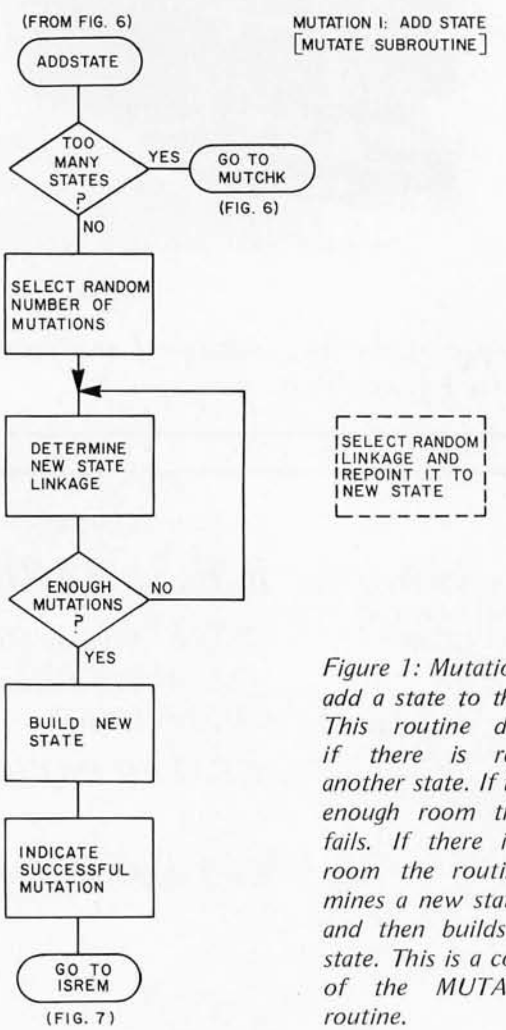


Figure 1: Mutation type 1, add a state to the model. This routine determines if there is room for another state. If there isn't enough room the model fails. If there is enough room the routine determines a new state linkage and then builds the new state. This is a component of the MUTATE subroutine.

Mutation 1 – Add a State

Figure 1 describes the first mutation type. As shown, the number of states is first compared to some maximum number. Most programs will have a fixed amount of memory allocated for containing the model, and so a check is made to see if any memory is yet available for expanding the model. The 2 symbol version previously discussed can conveniently be implemented using two bytes per state with the maximum number of states being 127. The programmer must determine the internal representation of the model and the amount of memory available and then set a variable to represent the maximum number of states the model can hold.

An Evolutionary Idea

The random number subroutine is called to provide an iteration counter. This counter is the number of randomly selected transitions that will be pointed at the new state. If no transitions were pointed to the new state then it would be an impossible state, ie: it could never be reached and could contribute no value to the model. Next, random transitions are changed to point to the new state which is to be created. Then, the new state is created. Finally ISREM is entered to remove any impossible states from the model.

Mutation 2 – Delete a State

Figure 2 describes the second mutation type. As with mutation type 1, a check is made of the number of states currently in the model. If there is only one state in the model, the mutation fails, since to delete the state would totally eradicate the model. If there is more than one state then the mutation will be able to proceed successfully.

Next, one of the states in the machine is randomly selected to be deleted. The only state that cannot be deleted with this subroutine is the state designated as the first state. To delete this state would result in the same mutation as mutation type 4 and is thus prohibited to this subroutine.

The actual deletion is accomplished in a rather roundabout manner. Every transition in the model is checked to see if it refers to the state being deleted. If so, it is repointed to some other random state number. The result of course is the creation of an impossible state, one that cannot be reached and is thus useless to the model. Entering the ISREM routine, as explained in the description of mutation type 1, will remove any impossible states from the model and so effect the actual deletion of the desired state.

Mutation 3 – Change a Transition

Figure 3 describes the third mutation type. The model is first checked for having more than one state. If there is only one state, then all transitions must necessarily point to that state, and so no change can be made and the mutation fails. Otherwise a

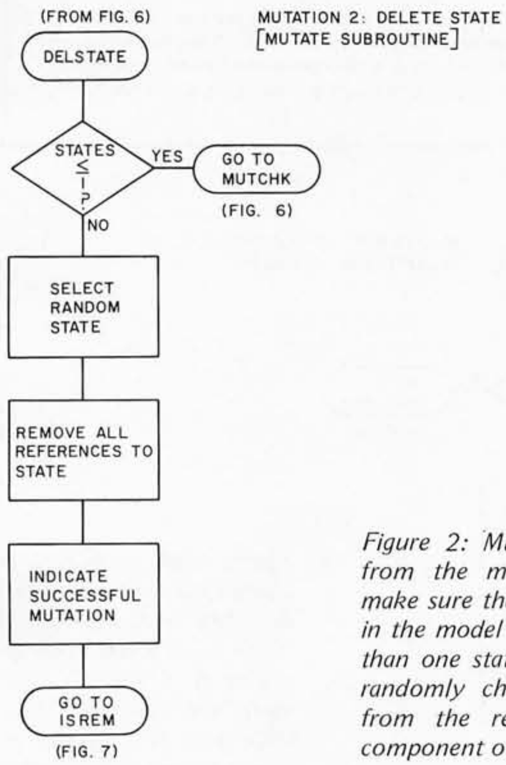


Figure 2: Mutation type 2, delete a state from the model. This routine checks to make sure that there is more than one state in the model or else it fails. If there is more than one state it deletes all references to a randomly chosen state thereby severing it from the rest of the model. This is a component of the MUTATE subroutine.

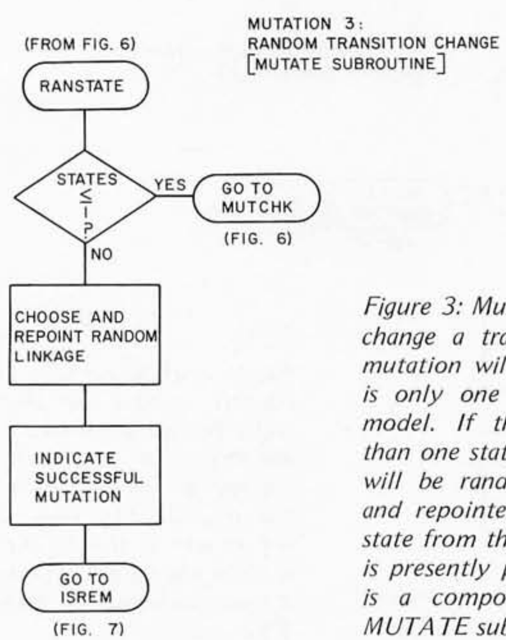


Figure 3: Mutation type 3, change a transition. This mutation will fail if there is only one state in the model. If there is more than one state a transition will be randomly chosen and repointed to another state from that to which it is presently pointing. This is a component of the MUTATE subroutine.

A Pseudorandom Number Algorithm

1. Test X, a 16 bit variable. If X is equal to zero, then set X to any number such as the time of day, or some other indeterminate number, and repeat step 1. This defines the initial seed of a pseudorandom sequence.
2. Multiply X by 259 to yield new value for X. Keep only the 16 least significant bits of the result. Increment X by 1.
3. If X is zero go to step 1.
4. Divide X by the input argument but do not destroy original value of X. The remainder of this result is a pseudorandom number between zero and one less than the input argument.

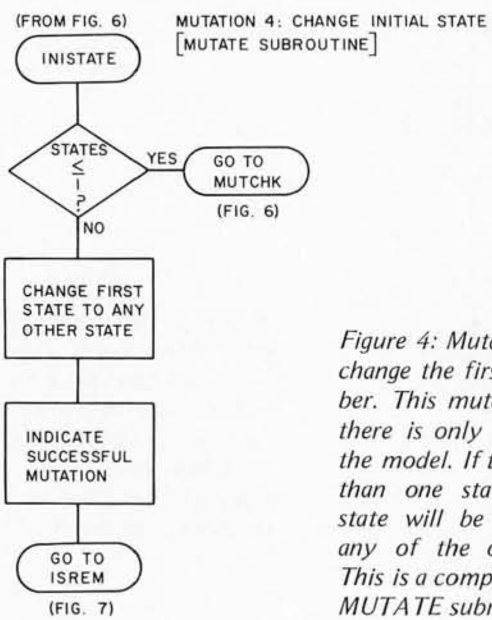


Figure 4: Mutation type 4, change the first state number. This mutation fails if there is only one state in the model. If there is more than one state the first state will be changed to any of the other states. This is a component of the MUTATE subroutine.

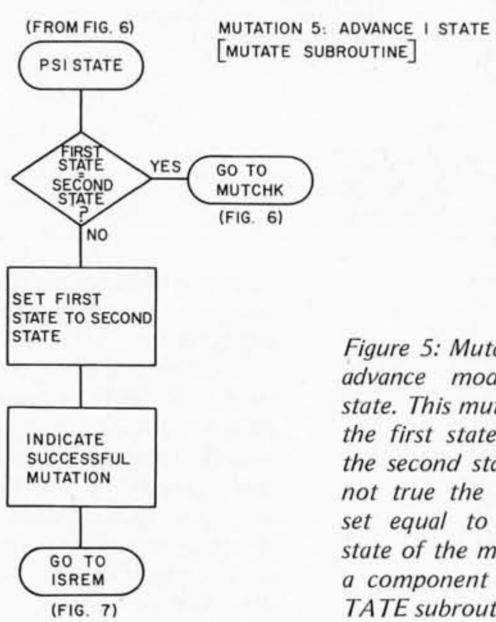


Figure 5: Mutation type 5, advance model by one state. This mutation fails if the first state is equal to the second state. If this is not true the first state is set equal to the second state of the model. This is a component of the MUTATE subroutine.

random element in the model is chosen and its transition pointer is changed to point to some other state. ISREM must be entered at the end of the routine since the changing of a transition may have resulted in the creation of an impossible state.

Mutation 4 – Change the First State Number

Figure 4 describes the fourth mutation type. Again, if there is only one state in the model, then it must necessarily also be the first state, and the mutation would then fail. Otherwise the first state number is set to any other state in the model. Also it is possible that an impossible state was created as the result of this mutation, so ISREM is entered to remove such impossible states.

Note that earlier discussion described the fourth mutation type as changing the current state number of the machine. The model must be driven by history from the first state when performing evolution, and it is seldom possible to work backwards from the current state number to determine the first state number. It is practical, therefore, to modify the mutation from change current state number to change first state number. The result of changing the first state number will usually end up changing the current state number.

Mutation 5 – Advance Model by One State

Figure 5 describes the simple but powerful mutation type 5. If the first state number is equal to the second state number of the model, then the mutation fails. Otherwise the first state number is set to the second state number.

Earlier discussion of mutation 5 described it as advancing the current state number by one. As with the case of mutation type 4, it is impractical to change the current state number directly, so the first state is advanced by one and evolution routines described later will run the model over its historical recall to result in the current state number actually being advanced by one state.

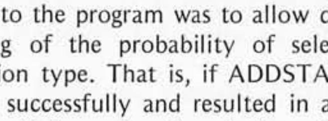
Mutation Control

Figure 6 shows the logic involved in controlling the mutation process. A random number is generated and used to select one of the five mutation types to be performed. Upon return from the mutation routine, a check is made for successful mutation. If the attempt was unsuccessful, a random selection is again made until a successful mutation is finally made. Finally, the routine OPTIMIZE is used to perform deterministic optimization and to assign a value to the model.

For those who implement this artificial intelligence game and would like to improve the intelligence forming process, the mutation control routine of figure 6 offers the greatest potential for improvement. The mutation selection process shown is simple and effective, but is essentially a brute force method. One of the first improvements I made to the program was to allow dynamic altering of the probability of selecting a mutation type. That is, if ADDSTATE was called successfully and resulted in a model with a higher value than before, then the probability of selecting ADDSTATE in the future was increased slightly. Similarly, if any mutation type was performed successfully and decreased the value of the model, then the probability of selecting that mutation type again was made slightly less. The most sophisticated versions of this artificial intelligence program use the very artificial intelligence process to optimize the selection of mutation types, thus aiding the evolution process.

ISREM – Remove Impossible States

Figure 7 describes the subroutine that removes impossible states. An impossible state is any state that cannot be reached. For example, the model:



f = X

has the impossible state Z. An impossible state cannot contribute to the value of a model but can hinder the model. If a model already had the maximum number of states permitted, but five of these states were impossible states, then the ADDSTATE mutation could not be performed until some states were deleted to make room.

The flowchart of figure 7 will not remove all impossible states. For instance, if in the figure above, the current state was changed from X to Z, then X and Y would now be impossible states instead of Z. The subroutine described will only find those states which are impossible solely because they cannot be reached from another state of the current model. In practice, this has never yet failed to eventually find all impossible states.

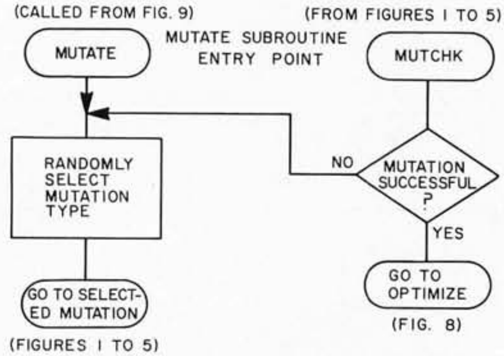


Figure 6: The MUTATE subroutine entry point. MUTATE chooses the type of mutation that will be performed using a pseudorandom number generator. At MUTCHK we check to see if the mutation was a success. If it was not then the MUTATE routine continues by choosing another mutation. (If at first you don't succeed . . .) If the mutation was a success then the optimization process cleans up the mutation by transforming it to its most usable form before returning control to the main program logic of PREDMUT.

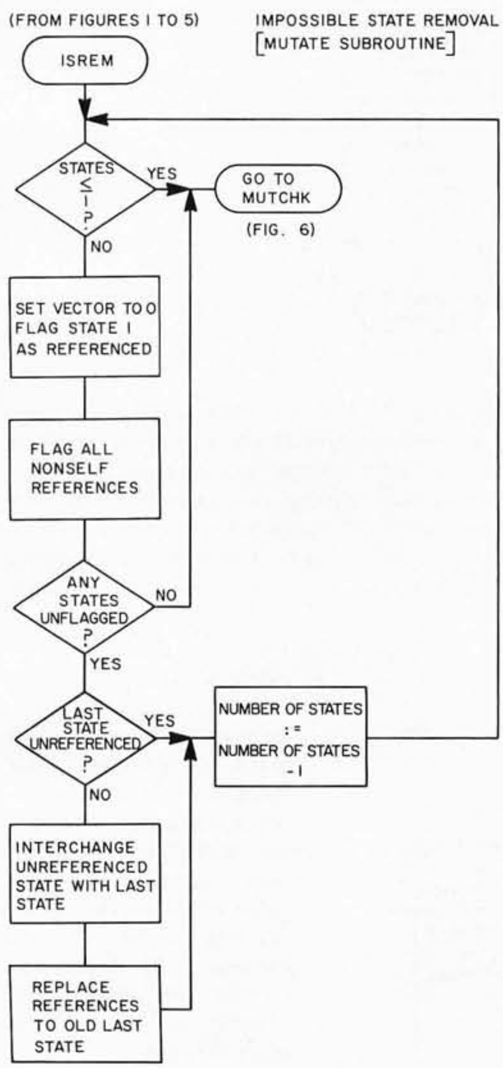


Figure 7: Routine ISREM removes any impossible states from the model. It performs this by checking all of the transitions to see that all of the states are referenced at least once. If it finds a state that is not referenced it will interchange that state with the last state of the model and delete the last position as valid. This is a housekeeping component of the MUTATE subroutine.

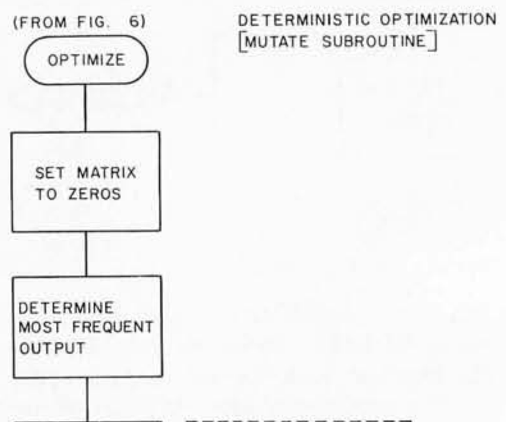


Figure 8: Routine OPTIMIZE determines the optimum prediction for each state of the new model. It will then evaluate the model using past history as a model to determine the value of the mutation. This is the final processing of the MUTATE subroutine of the program.

Table 1: The matrix used to evaluate the optimized form of the new mutation. Each predicted input symbol is matched against the actual input. The value for that combination is then added to the total value of the model. In this manner, using historical methods, the new model can be compared to the older model to see if it is more efficient.

prediction	actual occurrence			
	$symbol_1$	$symbol_2$	$symbol_3$	...
$symbol_1$	$value_{11}$	$value_{12}$	$value_{13}$	...
$symbol_2$	$value_{21}$	$value_{22}$	$value_{23}$	...
$symbol_3$	$value_{31}$	$value_{32}$	$value_{33}$	...
⋮	⋮	⋮	⋮	⋮

Looking at the flowchart then, the process is simple. First a table is set to zero. Every transition is then examined. If the head of the arrow points to a different state than the tail, then the entry in the table corresponding to the state pointed to by the head of the arrow is flagged.

After looking at all transitions, the table is examined. If any state in the model remains unflagged then it is an impossible state. If the impossible state is the last state in the model, then one need only decrement the variable denoting the number of states in the model to result in the deletion of the state. Otherwise the impossible state is first swapped with the last state in the model before the variable is decremented.

OPTIMIZE – Evaluate New Model

Figure 8 describes the most important segment of the program. OPTIMIZE performs the deterministic optimization mentioned previously, and evaluates the model with respect to the goal. Optimization and evaluation provide the criteria for the evolution selection procedure.

OPTIMIZE is conceptually simple, although often large of implementation. The model is made to perform with historical data. As each input observation is fetched and the appropriate transition made, the required output is noted. Thus for each transition we have a table of the number of times each output should have occurred for perfect prediction. The output symbol that should have occurred most frequently for a transition is then made the output for that transition. If the transition was never used during this historically driven run, then the transition is made to output the most frequently occurring output symbol over all of history.

After optimization, the model is rerun using the same historical data. Now, however, the optimized model is evaluated in terms of its ability to achieve the goal provided. The goal is expressed in a matrix of the form shown in table 1.

For each transition the prediction and actual occurrence are used to extract a value from the table. The sum of these values is the value of the model.

For the purpose of predicting primes, the matrix is defined in table 2a, while for earthquakes the matrix is defined as in table 2b. You see then that each accurate prediction of an actual earthquake or no earthquake adds 1000 to the value for the model. To predict an earthquake when one does not occur adds 500 to the value of the model,

If you talk computers, we speak your language.

1. MICROPROCESSORS Technology, Architecture and Applications
Daniel R. McGlynn

This book explains how a microcomputer system is built and used, and also provides a unique survey of the specific models of microprocessors available today. Gives useful actual applications including automotive, telecommunication, low-cost home computers, microprocessor selection criteria, microcomputer system design techniques, a semiconductor technology review, software and the use of development systems.

(0 471 58415-2) 1976
207 pp. $11.95

2. MICROPROCESSORS AND MICROCOMPUTERS
Branko Soucek

Here is a complete, detailed text and introduction to the field beginning with the basics and taking the reader all the way to programming. Some of the book's features include:

- Explanation of digital codes
- Simple hexadecimal, assembly, and high-level programming
- Logical systems and microcomputer organization
- Basics for design and use of microprocessor-oriented systems, and the replacement of powerful minicomputers with the microchip set.

(0 471 81391-5) 1976
607 pp. $23.00

3. COMPUTER-AIDED EXPERIMENTATION Interfacing to Minicomputers
Jules Finkel

Providing practical guidance on all major topics needed in interface specification and design, this book shows you how to connect scientific instruments and experiments to computers and to analyze suggested criteria for the selection of system elements. Offers many examples of practical connections and how to overcome problems that are commonly encountered.

(0 471 25884-9) 1975
422 pp. $28.00

4. INTRODUCTION TO MICROCOMPUTERS AND MICROPROCESSORS
Arpad Barna & Dan I. Porat

Presents a concise, basic introduction to microprocessors—what they are, how they work, and how to read the applications literature. You'll find descriptions of the basic structure of a microprocessor, arithmetic operations and circuits, basic programming techniques, and information on input/output, memory, assemblers, loaders, data structures, and subroutine linkages.

(0 471 05051-2) 1976
108 pp. $11.25

5. MINICOMPUTERS IN DATA PROCESSING AND SIMULATION
Branko Souček

Explores basic principles of digital codes and logical systems, concentrating on programming, organization and interfacing. Souček explains the elements of digital circuit design, the basic instruction set for a minicomputer, and data simulation necessary to design and use your digital systems.

(0 471 81390-7) 1972
467 pp. $24.50

WILEY-INTERSCIENCE
a division of John Wiley & Sons, Inc.
605 Third Avenue, New York, N.Y. 10016
In Canada:
22 Worcester Road, Rexdale, Ontario

6. A PRACTICAL GUIDE TO MINICOMPUTER APPLICATIONS
Edited by **Fred Coury**

Bringing together the work of leading experts, this volume provides information and examples of the application of minicomputer techniques, what is actually involved in the application and how to go about the selection and connection process. The five parts include an introduction to minicomputers, peripheral and software considerations, minicomputer selection, general applications and specific application.

(0 471 18051-3) 1972
211 pp. $6.45

7. SCIENTIFIC ANALYSIS ON THE POCKET CALCULATOR
Jon M. Smith

Smith shows, in clear step-by-step procedures, how to use and apply numerical techniques, approximations, tables, graphs, and flow charts for performing quick accurate calculations, all the way to advanced mathematics. Using few derivations and theorems, it's the best book for getting numerical results for scientific operations from your pocket calculator.

(0 471 79997-1) 1975
380 pp. $13.75

Please send the books indicated for 10-DAY FREE EXAMINATION. (Restricted to the continental U.S. and Canada.)

Mail to: WILEY-INTERSCIENCE
P.O. Box 092
Somerset, N.J. 08873

☐ Payment enclosed, plus sales tax. Wiley pays postage/handling. We normally ship within 10 days. If shipment cannot be made within 90 days, payment will be refunded.
☐ Bill me. ☐ Bill firm or institution.

☐ 1. McGlynn (0 471 58415-2)
☐ 2. Souček (0 471 81391-5)
☐ 3. Finkel (0 471 25884-9)
☐ 4. Barna/Porat (0 471 05051-2)
☐ 5. Souček (0 471 81390-7)
☐ 6. Coury (0 471 18051-3)
☐ 7. Smith (0 471 79997-1)

NAME ____________________

AFFILIATION ____________________

ADDRESS ____________________

CITY ____________ STATE/ZIP ____________

Prices subject to change without notice.

092 A 8071-57

(a)

prediction	actual	
	prime	not prime
prime	1	0
not prime	0	1

(b)

prediction	actual	
	earthquake	no earthquake
earthquake	1000	500
no earthquake	0	1000

Table 2: Two tables which illustrate examples of goals used to evaluate the models using historical data methods. Table 2a shows that when a correct guess of either prime or not prime is made a value of 1 is added to the total value of the state. Table 2b illustrates another manner of weighting answers. A correct answer receives a weight of 1000. Predicting an earthquake when there isn't one only adds a weight of 500 to the model. Not predicting an earthquake when there is going to be one adds nothing to the model. This is a more subtle form of weighting since it not only allows a good and bad answer but also a not so good answer.

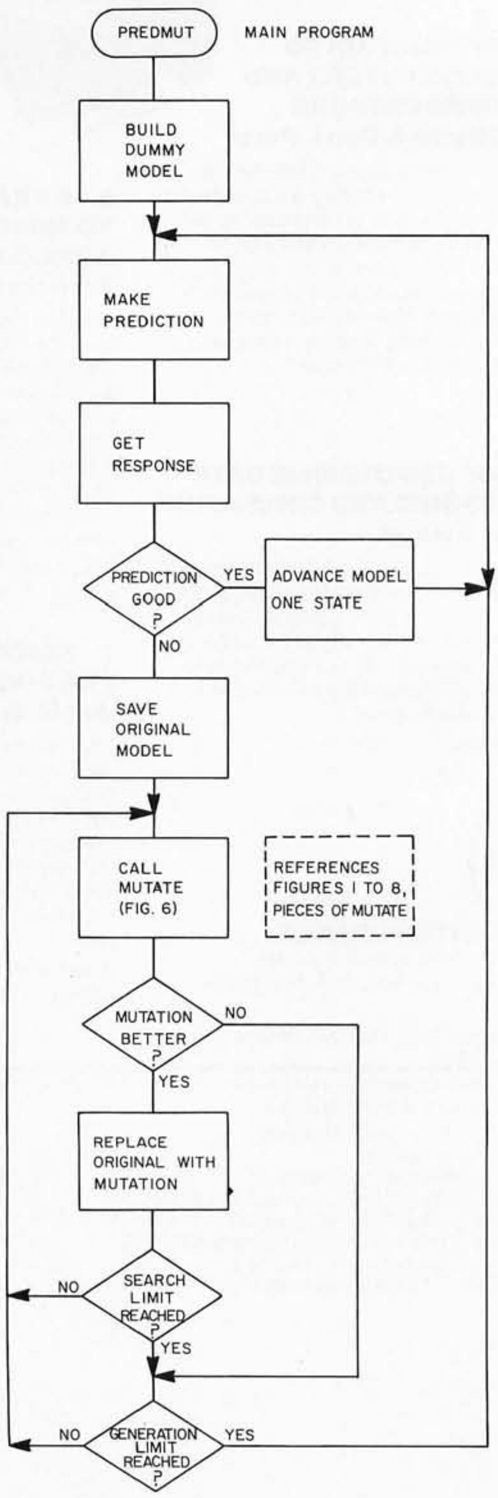


Figure 9: This is the main control logic of the simulated evolution technique. It first builds a dummy model, predicts an event, and receives the response. It then performs repeated versions and variations of the model looking for the optimum model to predict the correct answers.

while missing an earthquake prediction adds nothing to the value of the model.

PREDMUT – Putting It All Together

Figure 9 shows the mainline logic for the program. The flowchart should be obvious except perhaps for the use of the terms search limit and generation limit.

In order to prevent the evolution procedure from taking hours or days, it is constrained as to how long it can take. Each time an evolution cycle must occur, two variables are set. The search limit variable defines the number of times the parent may be serially mutated when searching for a better offspring. The generation limit variable defines the number of offspring to be generated.

The result is that a fixed number of offspring are generated. Further, each offspring can undergo a maximum series of mutations, since it is seldom that a single mutation results in a better model. To make the program more intelligent, one can increase these limit constants, but the result is a greater amount of computer time required between responses.

Again, for those adventurous programmers who want to make their programs even smarter, other methods can be employed to determine dynamically how many offspring and how many mutations are to be performed. There are other advanced evolutionary techniques that can also be employed. Interbreeding, majority logic and second order pattern recognition are just a few terms describing the advanced techniques available in computer science literature.

In Conclusion

If you have the patience, sit down with a pencil and paper and attempt to perform the process I've just described. Many mathematical books and papers contain random number tables you can use. As a result you should quickly see how this mathematical technique results in the creation of a model of some process. From there it is relatively simple to program.

Do not be afraid to try your own improvements, but do so intelligently as you get a feel for the underlying processes. The process described herein is the basis for many computer versions of this artificial intelligence technique and has worked with varying degrees of success for a wide variety of goals. I might conclude from personal experience: Be careful how and to whom you expose this technique. There are many people who fear computers as they fear anything they don't understand and their enthusiasm for your creation may not match yours. With a good enough program model, however, you should be able to predict who these people will be. [*Hmm . . . our artificial life has defense mechanisms.*] ■

Clubs and Newsletters

KIM Users Notes

Owners of KIM microprocessors should look into *KIM Users Notes*, published monthly by Eric C Rehnke, Apt 207, 7656 Broadview Rd, Parma OH 44134. The subscription rate is $5 for the first six issues.

Alberta Microprocessor

David Lavers, a member of the recently formed Alberta Microcomputer Society, has been in touch. Right now information at this end is a bit sketchy, but you can find out more through their acting president Dwight K Soloman, c/o The Computer Hobby Shop, 4812 16th St SW, Calgary Alberta CANADA T2T 4J5, (403) 243-6776.

Diablo Professional Users Group — Pleasant Hill CA

Here's a unique idea for a computer club.

The word professional in this case includes two categories: the computer novice who is an expert in another field, and the computer expert who can provide answers to the first group. This is an opportunity for budding technical types to get in on the ground floor of microcomputer consulting. The beginners include a wide spectrum of industries and professions. Anyone from the high school level up, with several years experience with hardware and/or software, can consider himself or herself an expert.

Meetings are held in the Diablo Valley College library's large conference room from 8:00 to 9:00 PM on the fourth Wednesday of each month. Diablo Valley College is located near the Willow Pass exit of Freeway 680. For details write or call Bob Hendrickson, Electronics Dept, Diablo Valley College, Pleasant Hill CA 94523, (415) 687-8373.

South Florida Computer Group

The South Florida Computer Group has apparently grown to the status of a major computer club. Meetings are held in both Miami and Ft Lauderdale. Meeting times seem so flexible that it would be best to contact them directly to get their schedule. Their newsletter *I/O* has improved substantially over the past six months to the point that it is far more than just a review and listing of club events. Volume 2, number 2 contains an in-depth look at resistors and capacitor color codes, and a piece on the new "minor loop" chips designed by Texas Instruments to be used with bubble memories.

To contact the South Florida Computer Group, write 1155 NW 14th St, POB 236188, Miami FL 33123, or phone (305) 324-5572.

Looking for Computer Games? Try POPULAR COMPUTING

POPULAR COMPUTING is a monthly compilation of computer games, puzzles and brain twisters. Most of the programs are fairly short, perfect for whiling away a spare afternoon. Among the games is up-to-date information on the micro field and a regular feature called "Schwartz on Calculators." To subscribe, write *POPULAR COMPUTING*, POB 272, Calabasas CA 91302. Subscription rate is $20.50 per year, or $17.50 if remittance accompanies the order. Add $1.50 per year for Canada and Mexico; all other countries, add $3.50 per year. Back issues are $2.50 each.

The Computer Hobbyist Group – North Texas

The Computer Hobbyist Group of North Texas is *the* place for computer buffs in the Dallas-Fort Worth area to develop and share an understanding of the hobby.

Meetings are held regularly at the University of Texas, Arlington, and the University of Texas, Dallas. Contact the Computer Hobbyist Group of North Texas at 2377 Dalworth 157, Grand Prairie TX 75050.

ON LINE – Hardware and Software Exchange

ON LINE is a newsletter strictly limited to classifieds for computer hobbyists. You'll find page after page of hardware and software available to buy, sell or swap. A great place to find a bargain or an odd item.

ON LINE is published every three weeks. Four issues are available for $1, 18 for $3.75 and 36 for $7. For advertising rates and other information, get in touch with D H Beetle, publisher, 24695 Santa Cruz Hwy, Los Gatos CA 95030.

Chicago Area Computer Hobbyist Exchange

Looking through *The Register*, newsletter of CACHE, I came across what may prove to be an interesting series on languages. In the coming months *The Register* is planning on a series of articles on BASIC, PL/I, SNOBOL, PASCAL, PILOT, FORTRAN, LISP, TRAC, CASUAL and FORTH. This should be a step toward clearing up language questions and misconceptions. Write to CACHE at POB 36, Vernon Hills IL 60061. *The Register* is available for $10 per year.

Washington Amateur Computer Society

The Washington Amateur Computer Society is a growing group of hobbyists in the nation's capitol. Meetings are held on the last Friday of the month in the second floor conference room of St John's Hall at the Catholic University of America. WACS is interested in exchanging newsletters with other computer groups. Correspondence should be addressed to Washington Amateur Computer Society, c/o 4201 Massachusetts Av, Washington DC 20016.■

Conducted by
Peter Travisano

BYTE's wide readership gives your club a unique opportunity to reach thousands of hobbyists. If you'd like to help your club develop or maintain a high profile, put us on your mailing list. Information about new clubs is especially welcome. Address your correspondence to Clubs and Newsletters, BYTE Publications Inc, Peterborough NH 03458.

Continued from page 74

blasted IV bytes. They are great if you never want to do anything fancy, but . . . .With the bus address coming from an extended microcode, one may simply decode a la 8080 and use 8212s as your IO ports. Also note that for some instructions there are free bits, ideal for adding instructions.

Interrupts will be painful, as the system doesn't have a stack, unless you use one of the aforementioned free bits to implement one (hint, hint).

My reaction to your suggestion about emulation of a 360 is unprintable. [*Quite reasonable evaluation . . .* CH] If you must, however, don't do it with the 8X300. Use Motorola's 10800 series. It's faster (50 ns, YES, 50 ns), and in the end would be less work. See, it's optimized for 360 type systems. Leave the 8X300 for the job it was designed for, a microcontroller.

Which gets around to the point of this letter: As a controller, the 8X300 is unbeatable. You neglected to mention what I consider to be the best feature of the system (after its speed). Every instruction operates on a bit string. You specify the starting bit and the length in bits of the field in the byte that you want the instruction to operate on. One can also do n bit rotates on any register to register instruction. Thus, you can add two registers and rotate right n bits in 250 ns or move bits 5 to 2 from memory to IV byte bits 7 to 4 without affecting the rest of the bits. This applies to all instructions. Thus, the 4 port kit can easily act as a front end to a busy computer for 8 Teletypes. Or consider a floppy disk. At double density, the bit transfer rate is 500 kHz, or a bit every 2 μs, or a bit every 8 instructions. Actually, as shift registers are so cheap, one might as well use them and free the microcontroller for other tasks like file handling. I have SMS's floppy disk controller interfaced to my 8080. At $640 it's not cheap, but they do provide the address and data bus on connectors for "maintenance." They are rather tight with the source code in the read only memories, though. It would be nice to have a parallel processing floating point processor when not doing disk transfers, particularly as it is practically free.

In short, the hobby world should take a long look at this device. But keep in mind that it was designed as a controller, not a computer. Its architecture is optimized as a bit banger, and it does that superbly. As an emulator, it's the pits: Use a 2901 and a 2909, or any of the other bit slices and save yourself a bunch of time, trouble and hassle. *Do* use the 8X300 as a peripheral controller,

floppy disk controller, unibus to Altair address mapper, DMA IEEE-448 bus interface, data acquisition controller, etc. If you use it for its intended purpose, you cannot be anything but delighted.

Finally, a note about Scientific Micro Systems. They are the only company to provide really complete documentation. Their products work as advertised. They gave me the shirt off their backs in helping with the nuclear instrumentation. It is truly refreshing to deal with such a company. I urge you to drop them a note as the microcontroller is only one of a large group of components they make. All are so pleasant to use, you kind of forget about them, which is the highest compliment I know how to give to an IC.■

A Critique of Self-Modifying Code

In February 1977 BYTE, page 132, we published an article by Charles Howerton, giving the software of a package of utility routines for an 8080 processor. Joseph Newcomer of Carnegie-Mellon University sends along this critique of the methods of coding the BARC routines. It should be noted that within the context of an interrupt free single process computer, the BARC routines will work as described. The subtleties occur in cases where interrupts are allowed or one desires to put the program into write protected or read only memory regions.

Joseph M Newcomer
Computer Science Dept
Carnegie-Mellon University
Pittsburgh PA 15213

Mr Howerton's article in the February BYTE would have performed a much greater service to the community of programmers if he had coded it to allow recursion and provided a few extra items of documentation. After having expended a great deal of effort and cleverness to fit his code into 256 bytes, he then compromises reliability by having it modify itself. The concept of code modifying itself is undoubtedly the worst single idea that neophyte programmers fall in love with. Since the most obvious thing in the world is to take his BARC package and put it in a PROM, my first critique is that he has carefully made sure that is not possible! Not only is self-modifying code useless in the world of read only memory, but experienced programmers know that code which modifies itself is inherently harder to debug.

I should like to offer the obvious solution: Instead of storing the dispatch address in line in the code, store it out of line in the writable programmable memory. Thus, one must reserve a certain number of bytes for this address. However, since none of the routines in BARC call one another, only one dispatch address area is required. Then one only need do a "call indirect" through this address. Of course, the 8080 does not have a call indirect instruction, so one has to fake it; use a 3 byte dispatch area, and make the first byte the op code of a JMP instruction. This is, of course, "modifying code," but at least we now have the ability to put the program in read only memory.

```
writeable space:
GODO      JMP       Q'000000'
rom, typical call:
          SHLD      G0D0+1
          . . .
          CALL      G0D0
```

Now, as to the issue of insufficient documentation: The functional documentation is quite clear, but the crucial piece of information that I find to be missing is that the routines in the BARC package should never be called from an interrupt service routine. This is because the routines are not reentrant (and my proposal of the dispatch vector in programmable memory does not change this); if they are active when an interrupt is taken, and the interrupt service calls them, the dispatch address will be changed. Consider the following sequence (events are listed):

```
CALL OCHR
. . .
CALL DASXR
. . .
CALL DASNT                                   }
-----<interrupt>-----                        }
. . . .                                      }
CALL XCHR                                    } Interrupt Changes
. . . .                                      } Dispatch Vector
<return from XCHR>                           }
-----<return from interrupt>-----            }
. . .
DASFC: CALL <something>
```

where <something> is expected to be OCHR but in fact is XCHR. The difference in time between the setting of the dispatch address and the use of it is (if I counted correctly) 235 machine cycles: a very, very long time. Furthermore, this sort of bug is nearly impossible to locate at the "lights and switches" level because the act of single-stepping changes the relative timings and the error will not occur. Using a debugging system may have the same effect, and if it is a homebrew debugger which would use BARC, then it would destroy the very information it was trying to analyze. Absolutely nowhere in the article could I find any warnings about this! Only my experience was an indicator: Whenever static data, either code or otherwise, contains state information, look for cases in which routines which use it can be called recursively. Note that recursion does not have to be explicit; calling a routine from interrupt level where the routine was active at the time the interrupt was taken constitutes a recursive call.

The use of a single dispatch vector does make it possible, with care, to use BARC from an interrupt routine. Before invoking any BARC routines, push the current dispatch address onto the stack; after using BARC, pop it off. If BARC was active, the program has correctly preserved state; if not active, only a few cycles have been lost.

The requirement that the parameter lists be in line also seems to be a bad choice to me; it would be much better to simply store a pointer to the parameter lists, or even better pass a pointer in a register. Since speed was not of the essence and size was, it seems to me that something which reduces the size of the code at the call site would be more desirable. Furthermore, if the addresses or values of the parameters must change, in line parameters force the user to write self-modifying code. This means that the user cannot, after having developed some neat system, convert it to a PROM region of memory.

A good criterion for evaluating a program is: Can it be put in PROM? If, after separating out the data areas, it cannot be put in PROM, then from my experiences I consider it to be badly written, no matter how useful or amazing its functions may be. Then ask: Can it be called recursively? If not, do not call it from an interrupt routine if it can ever be called from outside an interrupt routine. And don't forget multilevel priority interrupts, which are very easy to handle on 8080 architecture. Do not call the routine from outside an interrupt context if it can be called from within one, which is the complementary condition. Note that these are not always obvious errors to detect, since the interrupt routine may call something which calls something which eventually calls something else; and one day the "something else" is modified so that it calls one of the nonrecursive routines. The scenario is now set for disaster: Let an interrupt come in at the wrong time and you are set for a long, tedious and probably unrewarding debugging session attempting to locate a source of "random" behavior.

Charles Howerton Replies

In response to the critique of BARC by Mr Newcomer of Carnegie-Mellon University, I should like to begin by saying that I agree with his comments without equivocation for the environment which he hypothesizes. I also agree that the concept of modifying running code is very "hairy." However, even Mr Newcomer's proposed "fix" requires the modification of running code.

The BARC routines were written for newcomers to the computer programming field. It has been my experience that the greatest problems the beginner has in writing programs are manipulating the registers (hence, the register preservation facilities of BARC), and writing routines containing loops which are designed to perform the functions provided by BARC.

In addition, the newcomers to home computing or even industrial or business computing are rarely faced with programming an 8080 or Z-80 which is loaded with interrupt generating hardware. Those of us who work with interrupt driven hardware (and I do) would not use BARC for the simple reason that we are, presumably, sufficiently talented in programming arts that the creation of routines to perform equivalent functions is trivial.

As for embedding the parameters in line with the code, the one type of code that the average beginner writes well is straight line code with a few conditional jumps where decisions are required. Also, this is a fairly standard practice in large machine operating systems. Relative to the comment, "... if the addresses or values of the parameters must change ...", what is the difference whether they change in a work area or in the in line code? They must still be changed!

BARC was not designed to be called recursively; it was designed to be used as a programming tool by someone who was using straight line code to solve a problem. As for debugging it: it works.

Could BARC have been written to meet all of Mr Newcomer's criteria? Undoubtedly. However, not nearly as many functions could have been included in the same space.

Charles P Howerton
Digital Group Software Systems Inc
POB 1086
Arvada CO 80001 ■

Thanks to Charles Howerton For BARC

Thanks for the Howerton article in your February 1977 issue, page 132, "Add Some BARC to Your 8080." Now I can finally get rid of the needless duplication and generally sloppy coding of utility routines in my programs and turn the job over to BARC. What a great time and memory saver!

D M Bell
Vice President, Engineering
Handi Kup Company
195 Tamal Vista
Corte Madera CA 94925 ■

Introduction to Microprogramming

S M Quek
Stanford University
POB 9647
Stanford CA 94305

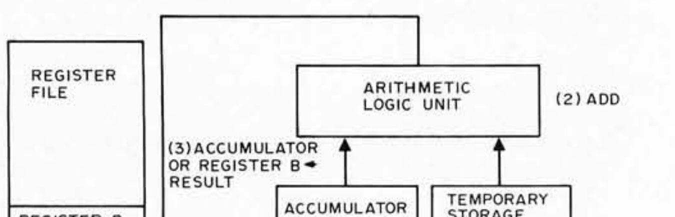


Figure 1: A block diagram with time notations for a sequence of events that might occur in a simple machine instruction such as: Add accumulator to register B. First (1) register B is put into a temporary storage area. Then (2) this storage area and the accumulator are added together by the arithmetic logic unit. The resulting answer (3) is stored in the accumulator or register B.

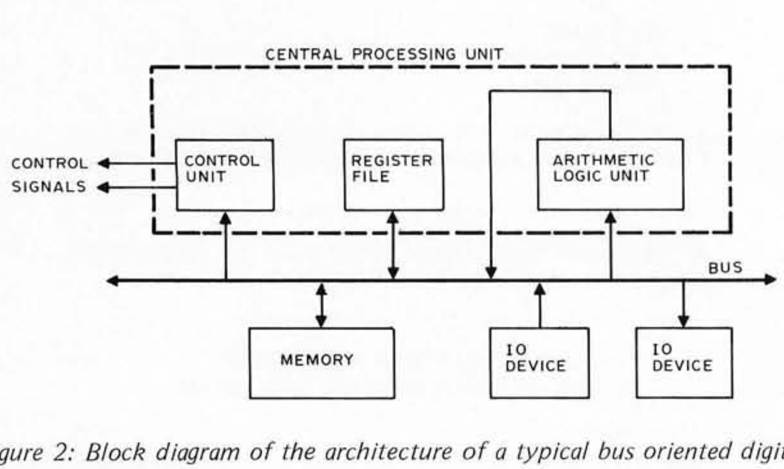


Figure 2: Block diagram of the architecture of a typical bus oriented digital computer.

What is Microprogramming?

When we consider the operation of a simple machine instruction, like add accumulator to register B on some computer, we often find that there is a sequence of even more elementary operations involved. For the example given, we may first have a transfer of data in register B to some temporary register in the arithmetic logic unit. Next, we may then perform an addition operation and finally, return the result of the operation to either the accumulator or register B. Figure 1 illustrates this sequence of operations.

Wilkes, an early pioneer in the field of computer design, called these elementary operations "microoperations." (See reference 1.) By this token, a single machine instruction, like the add described above, would consist of a microprogram of these microoperations. Microprogramming is, then, the implementing of control logic for a computer's instruction set through the ordered storage of processor control information.

Microprogrammable Computer Organization

Figure 2 is a simplified block diagram of the architecture of a digital computer. The organization of a microprogrammable computer differs from that of a nonmicroprogrammable computer in the design of the control unit. The nonmicroprogrammable machine uses a hardwired control unit. All control lines are fixed and cannot be changed easily. On the other hand, a microprogrammable machine uses a changeable microprogram in implementing the control unit and thus by changing the microprogram, the machine can be altered within certain limits of its design. Let us now take a look at the control unit of a microprogrammable computer and figure out how it works.

A typical microcontrol unit would consist of a mapper, a microsequencer, a microcontrol storage and a decoder. The last item, the decoder unit, is optional and may not be found in some machines. The interconnections between these units are shown for a typical design in figure 3.

In operation, a machine instruction is fetched from main memory and is stored in the instruction register. The mapper converts this machine instruction into the starting address of the microprogram routine which is supposed to execute the instruction as a sequence of microoperations. This address is passed on to the microsequencer whose job is to step through the microprogram. As each microinstruction is read out, the decoder translates it into control signals for the various control lines.

Originally, the mapper was implemented using a decoder tree made up of discrete logic gates. Nowadays, array logic blocks in the form of read only memory and so called programmed logic arrays are used for this purpose. (It should be noted that programmed logic arrays are especially suited for this task. Read only memories contain many more bits than are needed, and are thus more expensive than programmed logic arrays. The array is powerful enough to implement most functions needed and its lower cost makes it a very attractive candidate for the mapper.)

The function of the microsequencer is to provide a value for the next address of an instruction in the control memory. It can be thought of as having a microprogram counter and additional logic to test for conditional branches. Thus, in its simplest form, it could just be a presettable counter with associated circuitry for performing branches and conditional tests. However, most commercially available micro-

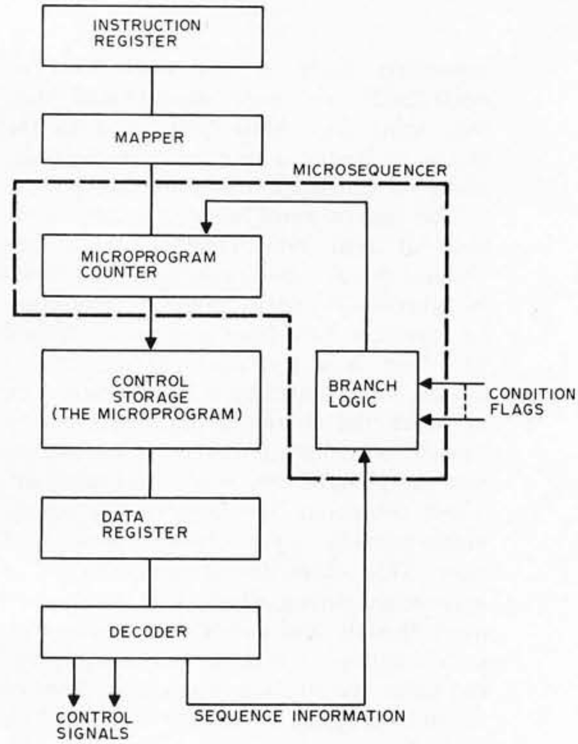


Figure 3: Block diagram of the connections between various parts of a microprogrammed central processor. The typical processor consists of a mapper, a microsequencer, control storage and an optional decoder.

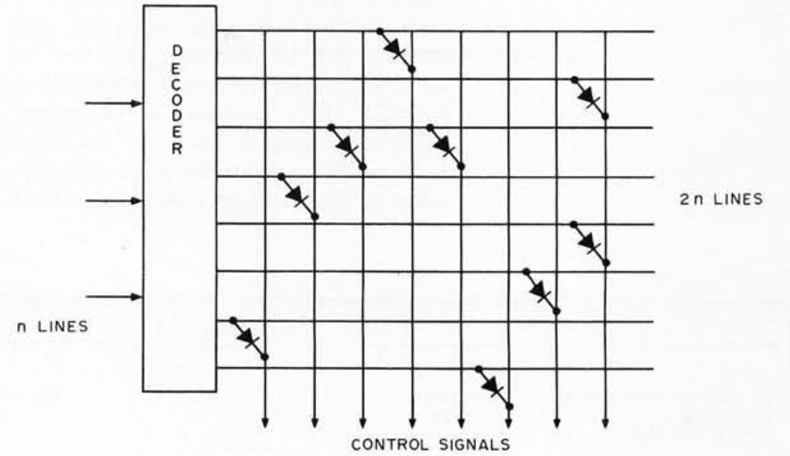


Figure 4: Typical diode matrix. A binary number is input on n number of lines. It is decoded into 2^n number of outputs. These lines are then decoded through the use of a diode matrix. The outputs of the matrix are at the control signals lines.

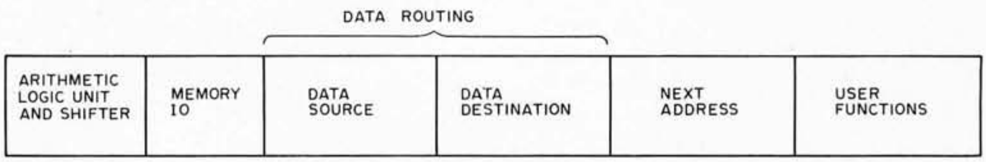


Figure 5: Typical microinstruction word format. The instruction allocations are generalized but are found in almost all such data formats. The word format itself can be of any length. The usual length is from 24 to 60 bits although a field width of 100 bits is used by IBM in some applications.

sequencers (such as the Intel 3001 or AMD 2909) are more sophisticated than this; some (eg: AMD 2909) even to the extent of having a built-in stack for processing microprogram subroutine linkages.

The miciocontrol store is usually some kind of read only memory. Wilkes envisioned this as a diode matrix such as shown in figure 4. Of course, immense technological advances have been made since Wilkes' time and now the microcontrol store is usually implemented by read only memories, of which the discrete diode matrix can be thought of as the forerunner. In addition to read only memories, some microprogrammable computers have a form of programmable memory as part of their microcontrol store. This allows for dynamic changes of microprogramming which lead to an even more flexible and powerful machine. With such a configuration, the microprogrammer can easily rewrite, add or delete portions of the microprogram to suit the particular task at hand.

From a consideration of the microcontrol store, we next proceed to a discussion of the microinstruction. It is the microinstruction that forms the control mechanism which causes each data register change. A typical microinstruction word format is shown in figure 5. Generally, there has to be a field to control the arithmetic logic unit and the shifter, one for memory, and one for IO control. In addition, another field has to be reserved for information regarding the routing of data. Some kind of sequencing field which specifies the next microprogram address is usually also included. Finally, to suit the architecture of a particular machine, there is a field left for user definable functions. These vary from machine to machine but would usually include conditional tests and branches. From this brief discussion, it should be apparent that there is no fixed width for the microinstruction. Indeed, it varies from 16 bits for the Signetics 8X300, chip, to 24 bits for the HP21MX minicomputer to 100 bits for various models of the IBM360. However, the width of the microprogram word in most small and medium size computers ranges from 24 bits to 60 bits.

If the microinstruction is wide enough, we can allocate a single bit to a single control line. In such a case, the microinstruction is said to be unpacked or horizontal. However, if we want to save control memory space, we may want to encode the data so as to compress the word width. An external decoder can then be used to recover the data. This is the packed or vertical format. A machine seldom takes on a fully packed or unpacked format for its microinstruction. Instead, most machines have microinstructions which lie somewhere between the two extremes; some fields are encoded while others are not.

When deciding on the width of the microinstruction, several factors have to be considered. The first and most obvious is cost: The wider the microinstruction, the higher its cost. This is so because memory is more expensive than a decoder. The penalty paid for having a vertical or packed format is a decrease in speed and flexibility. In the case of a horizontal machine, there are separate bits controlling the individual lines. Thus, there can be more parallelism in the control as more resources can be controlled simultaneously. However, for this same reason, horizontal machines are much harder to program. The microinstruction set of vertical machines resembles the assembly languages of minicomputers.

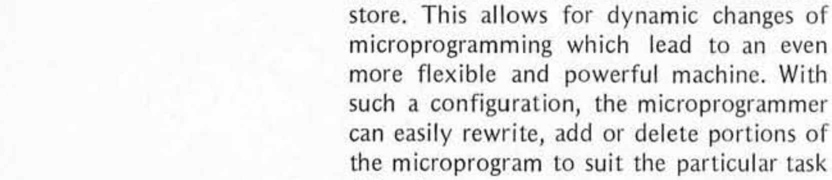


Figure 6: Timing diagram for the sequence of interpreting an instruction. This process is divided into five stages: instruction fetch, decode, data fetch, execution and testing. As soon as the testing is finished, the instruction fetch cycle is again encountered to start the next sequence.

Instruction Interpretation

Let us consider the sequence for the interpretation of an instruction. In figure 6, we see that the process of interpreting an

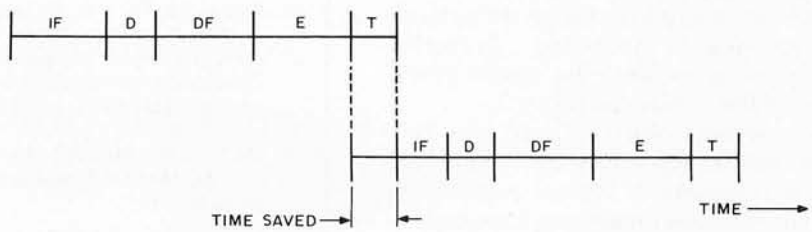


Figure 7: Timing diagram illustrating the time that is saved when implementing the changes of listing 2. This time saving occurs only when an interrupt or direct memory access request does not have to be processed. If a request does have to be processed, the timing diagram of figure 7 applies to the process..

instruction can be roughly divided into five stages. The first stage is that of an instruction fetch. The contents of the program counter is sent to the memory address register and a read memory command is initiated. The program counter is now incremented and the microcontrol unit waits until memory is ready with an instruction.

Once memory is ready, the instruction is loaded into the instruction register. This completes the first stage. On the time chart (figure 6), this corresponds to the segment IF. The mapper decodes this instruction into a microaddress which is passed on to the microsequencer. This is the decode phase, segment D on the time chart. As the microsequencer steps through the control memory, we have control signals coming out of the microcontrol memory. Depending on the instruction, we may need to initiate another memory read to fetch data. This would be segment DF. Once this is completed, we can proceed to instruction execution, segment E. Finally, upon completion of execution, a series of tests can be performed. These could include software tests for conditional branch hardware or software tests for interrupt, and hardware tests for direct memory access requests. The sequence for an add accumulator to register B (store results in accumulator) and skip if overflow may look like listing 1.

Of course the listing has to be coded into micromachine language form. DMA. SERVICE, INT.SERVICE and INSTRUCTION.FETCH would then be microprogram subroutines to service the various requests. A register, which holds data coming from memory, is assumed to be present. If no buffer register is used, then step 10 in the routine should be changed to:

(10) IR:= MEMORY DATA.

This will enable the mapper and load the microsequencer.

Why Microprogramming?

Before microprogramming was developed and firmly incorporated into computer design, most computer designers had to rely on multiphase "hardwired" logic for their design. Multiphase logic uses multiphase clocks to control the various register to register transfers and other functions. Hence, the designs are extremely complicated. Once the machine has been hardwired, it becomes virtually impossible to change the instruction set without redoing the design once again, ie: rewiring it.

Microprogramming overcomes these disadvantages and provides a means for obtaining relatively simple and flexible designs. To illustrate, reconsider the microprogram for the add instruction. If we look at the time chart of figure 6, we see that waiting for

Step	Instruction	Commentary
1	MAR:=PC	memory data counter:= program counter; [read memory]
2	PC:=PC+1	increment program counter;
3	[wait for memory];	
4	IR:=MDR	enable mapper, load microsequencer;
5	TEMP:=REG B	disable mapper;
6	ADD, ACC:=RESULTS	add and store results in accumulator;
7	IF OV=1 THEN PC:=PC+1	if overflow increment program counter;
8	IF DMA REQ=1 THEN JMP DMA.SERVICE	if direct memory access requested, go to routine;
9	IF INT REQ=1 THEN JMP INT.SERVICE	if interrupt requested, go to routine;
10	JMP INSTRUCTION.FETCH	fetch next instruction;

Listing 1: A program listing for the sequence: Add accumulator to register B, store results in accumulator and skip if overflow exists. Lines 1 through 4 are the instruction fetch routine. Lines 5 and 6 add register B to the accumulator and stores the result in register B. Line 7 checks for an overflow, and lines 8 through 10 check for interrupt and direct memory access requests. When the program listing is encoded into micromachine language form the DMA.SERVICE, INT.SERVICE, and INSTRUCTION.FETCH will become microprogram subroutines to service the various requests.

Step	Instruction	Commentary
9	MAR:=PC	read memory;
10	IF INT REQ=1 THEN JMP INT.SERVICE	if interrupt requested, go to routine;
11	JMP INSTRUCTION.FETCH+1	skip first step of instruction fetch;

Listing 2: Revisions of the program of listing 1 which allow a savings of time in the execution of the program. This savings is shown graphically in the timing diagram of figure 7.

memory to be ready with an instruction takes up a good portion of our time. Thus, to speed things, we may want to initiate a read memory instruction before we proceed to the various tests. Accordingly, we rewrite our microprogram with the modifications starting on line 9 shown in listing 2.

The new time chart obtained with this modification is shown in figure 7. Note that if no direct memory address requests or interrupts have been requested, then there is a savings in time. This is because we do not have to wait as long for the instruction fetch: a result of the early initiation of the read memory. If there has been an interrupt or direct memory access request, then no gain in speed would be obtained. However, since these requests are relatively rare, this new modification would result in an overall increase in speed.

If hardwired logic was used for the control unit, it would be very difficult to make the modification just described. Thus, we see that microprogramming is a very powerful tool in the design of digital computers.■

REFERENCES

1. Wilkes, M V and Stringer, J B, *Microprogramming and the Design of the Control Circuits in a Digital Computer*, Proceedings of the Cambridge Philosophical Society, Part 2, Volume 49, April 1953, pages 230–238.
2. Mick, J R, AM2900 *Bipolar Microprocessor Family*, Micro 8 Proceedings, September 1975, pages 56–63.
3. Coleman, V and Rallapalli, K, *A Versatile Microprogram Sequencer*, Micro 8 Proceedings, September 1975, pages 52–55.
4. Intel, "3001 Microprogram Control Unit," data sheets, Intel Corp, Santa Clara CA.
5. Cook, R W and Flynn, M J, *System Design of a Dynamic Microprocessor*, IEEE Transactions on Computers, C-19, 1970, pages 213–222.
6. Stone, H S, editor, *Introduction to Computer Architecture*, Chapter 10, "Interpretation, Microprogramming, and the Control of a Computer" by Flynn, M J, Science Research Associates Inc, Palo Alto CA, 1975, pages 432–471.

Your microcomputer is only as good as its memory

Your microcomputer is... in a manner of speaking... capable of thinking. The amount of information you can store, combined with the access speed, determines the bottom line capability of your microcomputer.

The Midwest Scientific Instruments' FD-8 Floppy Disk Memory System will enable you to get the most out of your microcomputer...6800 or 8080. We have the most complete 6800 disk operating system available today, which interfaces to any microcomputer via a single PIA chip.

When used with the MSI FDOS Operating System...included with purchase...the FD-8 gives your microcomputer a level of performance which can only be matched by larger computers.

You can have program save and load with named, variable length, password protected files at three levels: object code files, source code files, and BASIC files.

It has disk routines which can be used easily by low level assembler language programs as well as high level BASIC programs.

The coresident assembler/editor saves and loads source files from disk, assembles and stores object code on disk, and creates assembly listings on a high speed printer.

You get a full ANSI standard BASIC interpreter* with program load and save, from disk as well as tape, with both random and sequential disk data files. BASIC also has multi-I/O port capability under software control.

And all you need do to access any of these functions is type the appropriate commands on your terminal.

The FD-8 is complete, including all power supplies, cabinet, cables, software, manuals, and documentation.

You can order the FD-8 as a kit for $1,150.00, or wired and tested for $1,395.00.

Visit one of our dealers for a demonstration, or send for our free Catalog of MSI products.

American Microprocessors
Equipment & Supply Corp.
Chicagoland Airport, P.O. Box 515
Prairie View, Illinois 60069
(312) 634-0076

Microcomputer Systems, Inc.
144 South Dale Mabry Avenue
Tampa, Florida 33609
(813) 879-4301

Micro Store
634 South Central Expressway
Richardson, Texas 75080
(214) 231-1096

Vanguard Systems Corporation
6812 San Pedro
San Antonio, Texas 78216
(512) 828-0553

A-Vid Electronics Company
1655 East 28th Street
Long Beach, California 90806
(213) 426-5526

Electronics for Yachting
1525 South East 16th Street
Ft. Lauderdale, Florida 33316
(305) 525-3478
TWX 510-955-9484

High Technology
1020 West Wilshire Blvd.
Oklahoma City, Oklahoma 73116
(405) 842-2021

Computer Workshop
174 Ifield Road
London England
SW10 9AG
01-373-8571

Midwest Scientific Instruments

220 West Cedar • Olathe, Kansas 66061 • 913/764-3273
TWX 910 749 6403 (MSI OLAT)

☐ Before I forget...send me your catalog

Name ____________

Address ____________

City ____________

State ____________ Zip ________

BY577

*MSI BASIC is an additional $65.00

MSI is a dealer for Southwest Technical Products • Mastercharge & BankAmericard Orders accepted

220 West Cedar • Olathe, Kansas 66061 • 913/764-3273 • TWX 910 749 6403 (MSI OLAT)

Circle 119 on inquiry card.

Continued from page 13

ELIZA INTEREST

John Aurelius' letter (March 1977 BYTE, page 16) struck a resonant chord. When I succeed in getting my own system, I have plans to implement a version of the ELIZA program. ELIZA was first described by Weizenbaum in the *Communications of the ACM (CACM)*, Volume 9, Number 1, January 1966, and has subsequently been discussed by Weizenbaum and others too numerous to mention. ELIZA is still alive and productive, although not necessarily as a psychotherapist. I am referring to a version of ELIZA implemented by Shapiro and Kwasny (*CACM*, 1975, *18*, pages 459 to 462) as an interactive consultant for a timesharing system. ELIZA is a very general program whose specific personality derives from a script she is given. Shapiro and Kwasny reasoned that ELIZA would be a suitable mechanism for providing assistance to timesharing users. It was this application of ELIZA which intrigued me sufficiently to begin experimenting with her. Using their program as a point of departure, I have implemented my own (somewhat improved, I feel) version of ELIZA also designed to act as a timesharing system consultant. This program is now free of bugs (I think); however, the script is not yet very sophisticated, so a dialogue with the program is not yet rousing. I would be willing to provide a reasonably well documented listing to anyone seriously interested. Before the idly curious drop me a card requesting a copy, I should warn them that the program alone consists of 500 lines of SNOBOL 4 statements and requires 60 K words (the equivalent of approximately 360 K bytes) of storage to run on the Honeywell 635 computer for which it was written.

Now that everyone's enthusiasm has been dampened, I should admit that my choice of SNOBOL was expediency. ELIZA was originally written in LISP (also a memory hungry, slow, interpretive language). ELIZA requires the ability to search input text for certain desired content (pattern matching) and ELIZA's internal data structures are best represented as some form of list structure. These two attributes suggest that the most natural (but not necessarily most efficient) choice of a programming language is one that facilitates one or both of these ends. For implementing ELIZA on a small machine, I envision a relatively small core resident program to maintain the keylists and do the pattern matching, while the lengthy script is maintained as files on a secondary storage medium. These files would preferably be kept as random access files on a relatively fast device such as a floppy. A sequential device like cassettes might do, but I fear it would be intolerably slow. Unfortunately, ELIZA needs to keep a great deal of text at her fingertips (core resident in my SNOBOL program) for matching against input and for reconstructing into output. Again, I would gladly exchange ideas with anyone interested in implementing such a program.

Glen A Taylor
Wisconsin Research and
Development Center for
Cognitive Learning
University of Wisconsin
1025 W Johnson St
Madison WI 53706

COMMENTS ON TURING MACHINES

Jonathan K Millen's December 1976 article (page 114) on an actual hardware implementation of a universal Turing machine was very interesting. The relationship between the Turing concept of a universal machine (computer) and the capabilities of "real" computers continues to be ignored or misunderstood by many persons who think they know what a computer is. Millen's hardware project may help to enlighten these persons who view the Turing machine world as totally separate from the "real computer" world.

It may be of interest to note that Konrad Zuse appears to have investigated the Turing machine concept soon after World War II, hoping to find some ideas which could be used to simplify the construction of real computers. (My knowledge of this activity comes from a brief private conversation with Zuse at Los Alamos in June 1976.) He did not find any, of course, since we know now from experience that extremely simple machines in their primitive state are difficult to use in practice. Zuse's early work on computers was done without any knowledge of Turing's work, while it is known that von Neumann's input to early computers was not independent of such knowledge. (von Neumann became acquainted with Turing while Turing was at Princeton in 1937 and 1938 after the Turing machine work which was published in England.)

Millen makes some statements regarding the Busy Beaver Game that should be corrected. This game was invented by Tibor Rado and is described in the article "On Non-computable Functions" in the *Bell System Technical Journal*, Volume 41, May 1962. Millen has taken the liberty of adding an additional state to his Turing machines which he counts as a state for the Busy Beaver Game. This adds behavior that effectively eliminates his machine example from being considered in the game. It goes into a nonstop loop instead of halting. This also distorts the rules of the game. His 6 counting "4 state" machine is actually a 3 state machine by Rado's rules.

Millen then gives what he says are known results for 3 thru 7 states when he really should be saying 2 thru 6 states. In any case, the "known" results he mentions are still incorrect. The best results are as follows:

Current Results (1975) Busy Beaver Game

(Millen's States:)	Correct Number of States	Busy Beaver Score:	Determined by:
(3)	2	= 4	T Rado
(4)	3	= 6	S Lin
(5)	4	= 13	A Brady
(6)	5	$\geqslant 112$	D Lynn
(7)	6	$\geqslant 117$	A Brady from 5 state result of D Lynn
	7	$\geqslant 22{,}961$	M Green
	8	$\geqslant 3\cdot(7\cdot 3^{92}-1)/12$	M Green

While the summary I have shown here is in the process of being published, most of these results have been available in the open literature.

Allen H Brady
Univ of Nevada
Computing Center
POB 9068
Reno NV 89507

AUTHOR JONATHAN MILLEN REPLIES:

I am grateful to Dr Brady for his update on the Busy Beaver Game results. The terminal "copy" loop on state 4 of the example program was necessary because my universal Turing machine (UTM) has no automatic halt. A copy loop is an adequate substitute, because it is recognizable as such, and it does not change the contents of the tape.

Readers may be interested in the alternative Turing machine hardware realizations described in the following two references:

1. I Gilbert and J Cohen, "A Simple Hardware Model of a Turing Machine: Its Educational Use," *Proceedings of the ACM Annual Conference,* 1972.
2. Wakerly, J F, *Logic Design Projects Using Standard Integrated Circuits,* John Wiley and Sons, NY, 1976.

Jonathan Millen
66 Main St
Concord MA 01742

FURTHER SYS 8 EXTENSIONS

Readers who have been following the two recent articles in the January and February 1977 issues on improving SYS 8 Software Package 1 by Willard Nico may be interested to know about one of our products. This product is called Software Package 0.5.

This is a program in source code, plus a complete manual, for improving SYS 8 Package 1. In addition to the added commands and auto line capabilities discussed by Mr Nico, our program offers: insert, delete, and change string operations on a current line; string operations on a find first occurrence basis; page listing modes; reordering of line numbers; automatic tabbing; optional suppression of line numbers; and more. For the assembler, we add the following: octal numbers are accepted; a global symbol table for often used symbols; an ASC pseudo op for including real time output lines; output of the symbol tables; an expandable table of pseudo ops; and still more.

The program also adds the capability to assemble programs in sections as they are read in from a mass storage device. This means that program size is no longer limited by the amount of programmable memory available for source code files!

The source code translates to 1.75 K object code bytes. The user edits this in with his existing SYS 8 Software Package 1, conforming it to the limits of his system. For this gem of a program, we charge a mere pittance, a token $14.95.

Larry Weinstein
Objective Design Inc.
POB 20456
Tallahassee FL 32304

WINDMILL JOUSTING DEPARTMENT

I'm just getting started in this field and have enjoyed and benefited from the past six or seven issues of your publication. I will soon purchase and (I hope) have running a micro, both for personal use and as a fairly large data base device for a three man law office.

What prompts me to write this is the use of such expressions as "his/her" and "he or she" which appear with increasing regularity in your magazine. The zenith was reached in the February issue.

Let's be fair; you are leaving out a lot of readers. For example, there are certainly corporations and schools which own micros. Surely, the inclusion of only the masculine and feminine gender must seriously offend these neuter users. *[Let's start an "It" liberation front?]* Please, then, include "it" in such expressions as "Whenever the user finally gets his/her/its machine running . . . . "

Secondly, there no doubt are some partnerships or associations that own or use micros. Can you imagine the chagrin of the members of these bodies not to be included in all your pronouns! This will necessitate your saying: "Whenever the user finally gets his/her/its/their machine running . . . . "

Don't offend, for God's sake. And to heck with grammar or readability.

I am sure, if you really work at it, you could find even more ways to insure that your articles are hard to read — like, "owner/user," for example — we have to be accurate, right? Or "hobby/personal/recreation/small business" *every time* to describe (modify) the word "microcomputer." That'll really screw up the readers and give more money to the authors, too.

Be honest, folks; are you on some kind of a crusade? If not, please drop the abominal usage of English (even lawyers recognize the use of "his/her" and the like is bad) and get on with publishing valuable, concise and readable pieces for the benefit of your poor readers. If the gals are offended by the use of masculine pronouns, then fine — print 'em *all* in the feminine gender. Most men couldn't care less. But stop the foolishness, okay?

W C Welborn Jr
Caine and Welborn Law Office
2221 W Franklin St
Evansville IN 47712

Its, you? Gesundheit!

SOME COMMENTS ON MIKBUG

The following letter reaches readers in two parts. The main body of the letter is found here; the information in one paragraph of the letter is noted in "BYTE's Bugs" on page 160, and is not repeated here.

I was pleased to see John Rathkey's article "A MIKBUG Roadmap . . . " in February 1977 BYTE, page 96. The IO routines in this ROM are very useful and have saved me a lot of programming time.

The following comments and additions apply to table 1:

1. BADDR alters the contents of both A and B.
2. OUTHL and OUTHR both destroy the contents of A. Thus A must be reloaded if one desires to output both nybbles.
3. INHEX puts a hexadecimal digit in the right nybble of A.
4. OUT4HS outputs the four hexadecimal digits pointed to by X and X+1, then prints a space. X is incremented by 2 and the contents of A are lost.
5. OUT2HS also uses X as a pointer. X is incremented and A is altered.
6. OUTS is omitted from the list. This routine prints a space and begins at E0CC.

The routine labelled PSTR in listing 1 duplicates PDATA1 in MIKBUG. The only significant difference between the two routines is that PDATA1 uses the code 04 (EOT) to mark the end of the string rather than 00 as used by Rathkey.

The indexed mode JSR command can occasionally be used to save a few bytes of code when using the MIKBUG IO routines to and from the A register. Simply load the lowest address of the routines you will be using into X; the desired routines can then all be addressed with the 2 byte indexed mode JSR rather than the 3 byte extended mode. Of course, this technique won't help if you want to use the routines that require X as a pointer.

The MIKBUG program uses a block of programmable memory from A000 to A049. Since this is usually provided by a 128 by 8 RAM chip, the remaining 53 bytes from A04A to A07F are available to the user for data or short programs. In addition, space for a stack is provided from A014 to A042. Since many programs won't require such a large stack, some of these bytes can generally be used for other purposes. The stack pointer is initially set to A042 and increments downward as more stack bytes are required.

You have a fine magazine; keep up the good work.

D B Brumm
dB Engineering
224 Hecla St
Lake Linden MI 49945

NOTES ON ARTICLE CONTENTS

I have been reading BYTE for about a year now. I would like to congratulate you on your very interesting journal. However, it disturbs me that you continue to publish articles on programming and construction techniques only for the more well known microcomputer chips such as the 8080 and 6800. I contend that articles on microcomputers based on the 1802 and the 6502 would also be beneficial as they are also in popular use.

Leonard P Jacobs Jr
USF # 1570
Tampa FL 33620

We've had numerous articles on the 6502, already, starting with a review of the processor in November BYTE 1975 by Dan Fylstra, and continuing with several about the KIM-1 and its application or modification. In the near future we'll have David Brader's Komputar, a homebrew 6502 system plan. As for the 1802, or any other processor, what we print to a large extent depends on what people are doing, since the majority of BYTE articles are unsolicited contributions from readers actively engaged in experimentation. Based on recent data from readers, authors and manufacturers, there should be a bit of an upswing in 1802 awareness over the next few months.

THE EVOLVING LEXICON

A thorough answer to W Buchholz's question *[February 1977 BYTE, page 144]* about words that have passed from computer jargon into the general vocabulary would probably require a master's thesis.

The main reason is that several wholly new dictionaries have come on the US market in the past ten years or so. Among them are: *American Heritage Dictionary* (three editions), *Doubleday Dictionary, Random House Dictionary* (two editions). Several older dictionaries have been extensively revised, set in new type, or both: *Webster's New World Second College Edition, Merriam-Webster Seventh College Edition, Thorndike-Barnhart Advanced Dictionary*. And that's not all.

The *Oxford English Dictionary* has started work on a 3 volume supplement that will cover new words and meanings since about 1914. The first volume came out in 1972. The *Oxford English Dictionary* seems to be doing a more thorough job on computer terms than on general electronics! The entry for "control" in the new *Oxford English Dictionary* supplement quotes the 1948 MIT glossary and the 1955 glossary of the British Standards Institute. But the *Oxford English Dictionary* inexplicably skipped the electronic use of "emission" as the kind of signal (AM, FM, TV) a station sends out, although that word with that meaning has been around since at least 1927.

Almost the first thing I did when I got interested in microcomputers was to build my own glossary, starting with the lists in the back of the *IEEE Dictionary* (and acquiring other glossaries as I went). So I had a list I could check against a brand new dictionary (*American Heritage Dictionary*, College Edition of 1969) and a dictionary whose date of revision is known (*Webster's New World Dictionary*, Second College Edition of 1970).

American Heritage Dictionary started with a clean sheet. It has excellent typography but fewer entries than *Webster's New World Dictionary*. It uses larger type, and has very wide outer margins on each page where the artwork is put. Its vocabulary of computer terms includes:

accumulator, address, ALGOL, alphanumeric, analog computer;
base, bit; Boole, George; Boolean algebra;
chip, computer, computer language, converter ("a device that transforms information from one code to another");
data ("Numerical information in a form suitable for processing by computer"), *data processing, demodulation, digital computer;*
flip flop, FORTRAN;
gate ("a circuit extensively used in computers that has an output dependent on some function of its input");
hardware ("a computer and the associated physical equipment directly involved in the performance of communications or data processing functions");
information, information theory, input;
machine, machine language, memory, module, Murphy's Law;
PL/I, printer, print out (verb), *printout* (noun), *program* (noun and verb), *programmer;*
readout, real time;
software, storage.

Webster's New World Dictionary has more entries and smaller type. The College Edition is its number 2 product. Some of the computer definitions are eyebrow raisers:

accumulator, address ("the location in a computer's storage compartment of an item of information, identified by a number or other code"), *alphanumeric, analog, analog computer;*
bit, Boolean Algebra;
computer, console;
data processing, decoder, digital computer;
flipflop, FORTRAN ("a digital computer language similar to algebra");
gate;
hardware;
information, information theory, input;
language ("a special set of symbols, letters, numerals, rules, etc, used for the transmission of information, as in a computer"), *logic* ("the systematized interconnection of digital switching functions, circuits, or devices, as in electronic digital computers");
machine language, memory;
printer, printout (noun), *program* (noun and verb, two definitions each), *programmer* and *programer* (as shirttails, undefined, following program);
random access, read ("to obtain [information] from [punched cards, tape, etc]; said of a computer"), *readout, real time, routine;*
software, storage, store, symbolic logic;
throughput, track;
word, write ("to record information in a computer's memory or on a tape, etc, for use by a computer").

Webster's New World Dictionary had a contributing editor who was then the head of the Electronics Engineering Department at Carnegie-Mellon University. *American Heritage Dictionary* had no electronics specialist identified as such on its 1969 masthead.

If you know someone who is majoring in computer and minoring in linguistics, you might set him or her on this. The sooner it's wrapped up thoroughly, the easier it will be to do thoroughly. One question that should be explored in any exhaustive treatment is why certain words were admitted to the general vocabulary, and others were not, in each edition of each dictionary.

C J Mike Fern Jr, WA6OWJ
1046 S Westlake #1
Los Angeles CA 90006■

BYTE's Bits

A Phonograph Record in a Computer Magazine?

Bob Jones of *Interface Age* magazine has informed us of an intriguing new feature included in their May issue: a flexible plastic phonograph record bound into the magazine which contains a 4 K byte BASIC compiler plus a binary loader and two memory test programs. The program, written by Robert Uiterwyk for Motorola 6800-based systems, uses the 300 bps Kansas City standard. The record's appearance marks the first time that such a technique has been used in a magazine to disseminate software. (Readers may recognize the record as the type used to promote recordings by mail).

To recover the programs, the user plays the record on his or her phonograph in the normal manner and feeds the output to an AC-30 or similar cassette interface, the same procedure used for data cassettes. Using the "tape out" feature (available on most amplifiers) is probably the easiest way to feed the signal to the interface.

Each record is good for about 100 playings; unless they are severe, scratches have no effect on accuracy. An additional benefit of this system is the elimination of tape dropout problems. Bob has promised some 8080 and Z-80 programs for future issues. Contact *Interface Age,* POB 1234, Cerritos CA 90701.■

KIM Has a Contest

In a press release from MOS Technology Inc, Richard Simpson has announced the KIM Software Contest, open to all KIM owners and users. The prizes are:

First prize: KIM-3 8 K memory expansion board
Second prize: KIMROM-1 Resident Editor/Assembler ROM Set
Third prize through tenth prize: KIMath Source Listing and User Manual.

All entries must contain program documentation and source code listing (but a hand assembled source is allowed). All entries become the property of MOS Technology Inc and will be turned over to the KIM Users Group for possible publication.

Entries will be judged on the basis of originality and usefulness to the user community. If external hardware is required, a schematic should be provided. Complex programs taking more than 1 K bytes of memory such as high level languages, assemblers, cross assemblers, text editors, etc, will be awarded a duplicate first prize if accompanied by working source tape or cassette. All entries must be received by July 1977. Prizes will be awarded on August 1 1977. Send all entries to KIM Software Contest, MOS Technology, 950 Rittenhouse Rd, Norristown PA 19401.

This looks like an interesting opportunity for KIM enthusiasts to earn a bit of glory. ■

The Following Was Received From American Federation of Information Processing Societies (AFIPS):

The fast growing field of personal computing will be in the national spotlight this June at the 1977 National Computer Conference in the Dallas Convention Center. Reflecting the dynamic growth and promise of the personal computing field, the 1977 NCC will feature a Personal Computing Fair, a Personal Computing Exposition, two full days of program sessions, a National Club Congress, as well as special interest sessions for computer hobbyists.

The Personal Computing Fair, scheduled to run throughout the four days of the conference, June 13 to 16, will feature operational displays and demonstrations of individually and group owned noncommercial projects. More than 100 small computing systems are expected to be displayed featuring hardware and software implementations, games, recreation, music, art, amateur radio, as well as scientific and general applications. Prizes and awards will be presented in recognition of outstanding achievement. The Personal Computing Fair will provide hobbyists with the unique opportunity to obtain new ideas for their own systems, solutions to current problems, and a wealth of "how-to" tips on personal computing.

The conference program will feature an in-depth examination of personal computing on Wednesday and Thursday, June 15 to 16. Two 3 hour panel sessions on Wednesday will examine "Personal Computing — Past, Present and Future" and "Hardware for the Computer Hobby Market." Thursday morning will feature a 3 hour panel covering "Personal Computing Software," with the afternoon devoted to the presentation of papers relating to personal computing, plus a concluding panel on "The Future of Retail Computer Stores." Each panel will feature presentations by leading authorities in the personal computing field, and will be designed to

THE COMPUTER SHOP

Calgary, Alberta
3515 18th St SW

Phone (403) 243-0301

IMSAI, SWTPC, Digital Group, Cromemco, T.D.L., Lear Siegler ADM-3A, Morrow cassette interfaces, Mini-Term, Polymorphic, Solid State Music, and the Calgary made Interalia 8000 micro. Prices generally US list plus duty, tax & exchange (now 30% altogether). Special: free Morrow serial/parallel/cassette port board incl. PROM boot with every IMSAI 8080 mainframe — first 50 orders mentioning this ad only! Full repair facilities.

CANADA

Circle 179 on inquiry card.

LOGIC DESIGN INC

QUALITY Support for those who wish to dig deeply into digital design.

Breadboarding System, textbook on logic design, and complete schematics for an elegant CPU that will run the PDP8 instruction set .. $1095

132 IC's .... $ 87

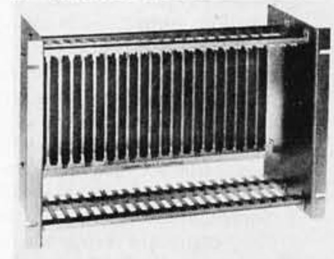
TTY Interface .... $ 20

Memory $26/k

BOX 3991, UNIV. STATION
LARAMIE, WY 82071
(307) 742-7977

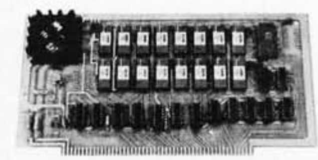
Circle 209 on inquiry card.

19" RACK S-100 BUS CARD CAGE



ECT-100 CARD CAGE KIT $100
CARD CAGE / MICROCOMPUTERS
ECT-100-8080 KIT $320
ECT-100-Z80 KIT $420

8,192 x 8 BIT STATIC MEMORY
EXCEPTIONALLY LOW POWER



8KM MEMORY BOARD KIT $295
PB-1 PROTOTYPING BOARD $28
EXT-100 EXTENDER CARD $24

ELECTRONIC CONTROL TECHNOLOGY
P.O. Box 6, Union, New Jersey 07083
(201) 686-8080

Circle 47 on inquiry card.

BYTE Article Index

FREE

Complete listing of all feature articles appearing in Volume I of BYTE— September, 1975 thru December 1976. Indexed for easy reference. Includes all errata.

To get yours, send a $.24 stamped self addressed envelope to:

BYTE Index
70 Main Street
Peterborough NH 03458

BYTESHOP™

the affordable computer store

7825 BIRD ROAD
(305) 264-2983
DIAL 264-BYTE

OF MIAMI

WE HELP YOU GET YOUR SYSTEM UP AND RUNNING.

The word is getting around . . . the Byte Shop **OF MIAMI** is a remarkable exception to the rule among computer stores. We offer a truly delightful environment supported by

- REAL courtesy
- REAL expertise
- EXPERT service
- GREAT classes

demo systems

IMSAI 8080	MEMORY EXPANSION
BYTE-8	COLOR TV GRAPHICS
SWTP MP68	LEAR SIEGLER ADM 3
CROMEMCO	PAPER TAPE READER
PROCESSOR TECH	

INTERFACES (KITS or ASSEMBLED UNITS)

Opening COMPUTER STORE
#2
IN FORT LAUDERDALE

Circle 271 on inquiry card.

DISCOUNTS!

IMSAI 8080	KIT $599.
assembled	750.
IMSAI 8080 w/22 slot MB	645.
assembled	795.
RAM 4A-4	125.
assembled	225.
MULTIPLE I/O Board (MIO) (both Tarbell and Byte modes one serial, two parallel, one control port)	175.
assembled	275.
EXPM (100 pin w/edge conn	6.50
assembled	11.50
(POLYMORPHIC SYSTEMS) 16x64 CHAR VIDEO BOARD	189.
assembled	229.
Keyboard — 63 key fully encoded ASCII, +5v only, positive or negative logic, repeat key, IC sockets.	kit $54.
assembled	69.

*write for discount prices on other IMSAI products.

*N.C. residents add 3% state sales tax

COMPUTER ELECTRONICS
BOX 339
Cary, N.C. 27511

Circle 253 on inquiry card.

provide attendees with the latest information on new developments, trends, and the outlook for the future. Ample time will be allotted to answering questions from those in the audience.

Plans are also under way to bring together various special interest groups in personal computing for a series of informal sessions on such topics as the building of computing kits, debugging software, use of assembly language, peripheral interfaces, cassette and disk storage, and software standards. In addition, plans are being developed for a "National Club Congress" to enable representatives of clubs from throughout the nation to exchange ideas and discuss issues relating to their activities and programs. Among expected topics will be whether or not a national personal computing association is needed, and if so, how it might be formed. Related topics are expected to include hardware/software standards, a possible national program library interchange, and the establishment of educational seminars.

In addition, the 1977 NCC will feature a commercial exhibition by equipment manufacturers and suppliers of personal computing products and services. The Personal Computing Exposition will be in the North Hall of the Dallas Convention Center, one level below the main NCC exhibit hall.

Information on the 1977 NCC may be obtained from AFIPS Headquarters, 210 Summit Av, Montvale NJ 07645, or by calling (201) 391-9810.■

A Calgary, Alberta Store . . .

The Computer Shop is the name of a new store which sells IMSAI, Digital Group, Interalia, Cromemco, Lear-Seigler, Polymorphic Systems, Southwest Technical Products, Morrow, Mini-Term and TDL products to central Canadians. The shop sent us a flier, with a handwritten note that the typical prices are USA prices plus about 25%. The store is located at 3515 18th St SW, Calgary, Alberta CANADA T2T 4T9.■

Another Dallas Area Store

KA Electronic Sales, a Dallas distributor of industrial electronic components to both businesses and individuals, has opened a computer store at 1220 Majesty Dr in the Brookhollow Industrial Pk, Dallas TX.

The KA Computer Store currently markets central processing units and peripherals by several manufacturers including IMSAI, Southwest Technical Products Corporation, Lear-Siegler Terminals, Solid-State Music and The Digital Group.

KA also supplies electronic components and parts as an industrial and retail distributor, and has a second electronics parts walk-in store located at 1117 S Jupiter, Garland TX.■

How to Get a BASIC Source Listing

Dr Dobb's Journal of Computer Calisthenics & Orthodontia, in its January edition of this year, has published the complete source and object code assembly listing of the Lawrence Livermore Laboratory BASIC interpreter developed by John Dickenson, Jerry Barber, John Teeter and Eugene Fisher. The interpreter is a 5 K byte program designed to be loaded in PROM or ROM. It includes a floating point arithmetic package. *Dr Dobb's* is located at People's Computer Company, POB 310, Menlo Park CA 94025. ■

Question:

Dr Chuck Adams of the Texas A & M University EE Department posed the following question in a recent phone conversation: "Who invented the D flip flop?" Can a reader supply the answer to this query, for publication in a future issue?■

Want to Find Out Who's a Professional Computer Scientist?

The 1977 Association for Computing Machinery Roster of Members, an alphabetic and geographical cross-listing of the names and addresses of more than 35,000 ACM members as of January 1 1977 is now available.

The Roster may be ordered from the ACM Order Department, POB 12105, Church St Station, New York NY 10249. Prices are $7 to members and $25 to nonmembers, prepaid.■

Guide to Buzzwords

"Sherry's Guide to Data Communications Buzzwords" is the name of a 24 page booklet of words and definitions which are commonly used in the data communications field. Write for your complementary copy, available from: Public Relations Dept, International Communications Corp, 8600 NW 41st St, Miami FL 33166.■

Survey Sweepstakes Results . . .

In November 1976 BYTE ran a random survey of readers, to gather data for editorial and marketing purposes about this crazy field. Of approximately 2100 survey questionnaires mailed, 1448 were returned prior to the deadline of November 15 1976. As an incentive to return the survey, we offered five Life Subscriptions to BYTE, commencing with the expiration of the current subscriptions of the winners. The following five individuals were drawn at random from the returned survey sweepstakes entry blanks (which were kept separate from the actual questionnaires in order to keep the questionnaires anonymous and private).

Arthur H Bazell
50 El Camino Real
Berkeley CA 94705

Allen L Curl
Robert S Curl & Assoc
1555 Alum Creek Dr
Columbus OH 43209

Dennis A Hewitt
POB 8747
S Charleston WV 25303

Mark T Marshall
18229 Topham St
Reseda CA 91335

Howard Rothman
86-25 Van Wyck Expy
Briarwood NY 11435■

Technology Fact
Attention ~~Science Fiction~~ Lovers . . .
Another Far-Out Technology

How About Running a Real World Enterprise Instead of a Computer Driven Simulation Game?

A group of aerospace specialists has begun to investigate the prospects for a satellite launch center at the equator. Sponsored by the Sabre Foundation, the group hopes to determine the extent of interest by government and private organizations in an "Earthport" that would be open to peaceful users from every nation.

Equatorial sites offer cost savings for most satellite launches because the earth's spin gives rockets a boost into orbit. Several nations now operate equatorial launch sites of varying sizes, but none are international.

"In the past six years, aerospace companies such as Boeing and General Dynamics have explored the possibility of providing commercial launches from the equator," said the director of the study, Mark Frazier. "We plan to work with representatives of private organizations as well as governments to determine what environment would be best suited for them."

The initial stage of the study is intended to assess international interest in establishing a "space freeport," and will be completed within the next four months, according to Frazier.

Project members will then evaluate the economic, technical, legal and political aspects of establishing an international launch site, culminating in publication of a report late this year. An international advisory group will review the report before any moves to approach Third World nations about hosting the spaceport.

Total cost of the study is estimated at $50,000, to be provided by contributions from philanthropic, space oriented organizations and individuals.

Although the project is at an early stage, it has drawn support from leading aerospace figures. Among the advisors are Prof Freeman Dyson of the Institute for Advanced Studies, Princeton; Dr Philip K Chapman, a former astronaut; Dr Raymond Bisplinghoff, past NASA associate director and research director; and Dr George Robinson of the Smithsonian Institution.

Three study groups will be responsible for preparation of the final report. The Government Launch Activities Committee will examine ways that governmental organizations could use an equatorial spaceport to their advantage, and the Private Users Committee will investigate opportunities for nongovernmental groups. The Freeport Design Committee, chaired by Stanford freeport specialist Dr Avlin Rabushka, will recommend possible sites and alternative legal and economic configurations of the site.

Copies of an eight page brochure describing Earthport are available for $.50 per copy. Suggestions about the project are welcomed by the foundation. For further information, write Mark Frazier, Space Freeport Project, Sabre Foundation, 221 W Carrillo St, Santa Barbara CA 93101.

Project Members
(Partial Listing)

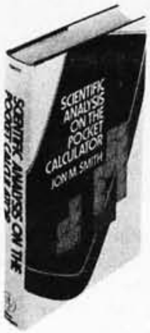
Advisory Board: Dr George Robinson, Smithsonian Institution; Dr Philip Chapman, former astronaut; Prof Freeman Dyson, Institute for Advanced Studies, Princeton; Pat Gunkel, Hudson Institute; Dr Larry Smarr, Harvard Astrophysics Center; Prof Alvin Rabushka, Hoover Institution, Stanford; Dr Raymond Bisplinghoff, former NASA associate director, NASA research director, and dean of engineering at MIT; Robert Prehoda, consultant.

Study Groups. Government Launch Activities Committee: Arthur M Dula JD, chairman. Private Users Committee: Robert W Poole Jr, chairman; Raymond L Kendall, program development manager, Motorola Inc; Paul Siegler, president, Earth/Space Inc. Freeport Design Committee: Prof Alvin Rabushka, chairman; Michael Bader, assistant director, NASA-Ames; Jerry Glenn, consultant.

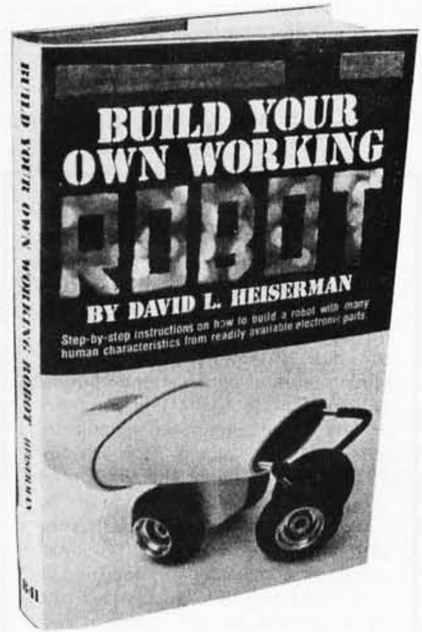
Executive Director: Mark Frazier.■

BITS™

___**Game Playing With Computers** by Donald D Spencer, published by Hayden. What does it mean to play games using a computer? Read this book to get an introduction into numerous recreational uses of the computer to program and play mathematical and logical games. Topics include numerous mathematical problems, casino games, board games, unusual gambling games, and miscellaneous logic games. Numerous BASIC language programs and listings are included to show details. $16.95.

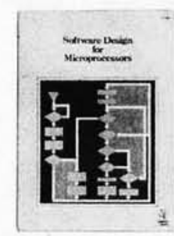


___**Scientific Analysis on the Pocket Calculator** by Jon M Smith, published by John Wiley & Sons. This book is another in a set of source books for mathematical analysis using the contemporary products of technology. It is oriented to the pocket calculator, yet it will provide you with algorithms and methods useful with any personal computer which implements the scientific and analytical functions found on a good pocket calculator. For a more complete description, see the book review on page 120 of the December 1976 BYTE; or order its 392 pages of detailed technical information and review its use for yourself. $13.75.



___**Build Your Own Working Robot** by David L Heiserman, published by Tab Books. This book will not tell you how to build Robbie, the robot of **Forbidden Planet**, or a classical android of science fiction. What it will introduce you to is the problems of making a robot mobile device called Buster III, using pre-microprocessor TTL integrated circuits for all logic functions. It is a must book for background reading, but much of the logic can be extremely simplified using today's microprocessor technology. Use this book as a first look at these problems from which you can build further and more elaborate solutions. Softbound, $5.95.

___**A Dictionary of Microcomputing** by Philip E Burton. In the opinion of BYTE's editor, Carl Helmers, "This is one of the best *designed* and executed dictionaries of computer related terms yet seen on the market. It is of particular relevance to those individuals who want a good general reference to numerous technical terms, broadly covering hardware and software fields as currently practiced." This new hardbound edition is part of the Garland Reference Library of Science and Technology. $12.50.



___**Software Design for Microprocessors.** This stand alone guide to microprocessors has been designed by the people at Texas Instruments to convey knowledge to the first time user of microprocessors. This excellent source book of computer concepts begins with an outline of the basic principles of the general purpose computer, its machine architecture, software, and methods of addressing. It proceeds to discuss how to build software, what is involved in documenting what you've done once you've done it, the mechanics of programming, and specific examples using the TI TMS-1000, TMS-8080, TMS-9900 and SBP0400 designs. You'll find a thick hardcover textbook filled with over 370 pages of useful information including a comprehensive glossary of microprocessor terminology, among several other detailed appendices. $12.95.

Send to:

BITS, Inc
70 Main St
Peterborough NH 03458

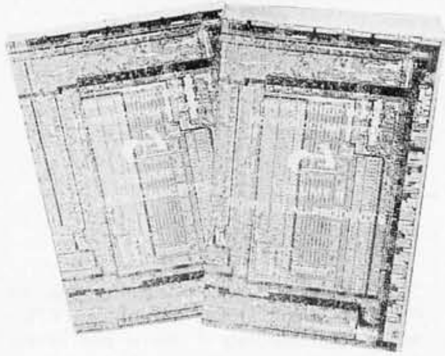
Check payment method:
___My check is enclosed
___Bill my MC No. ____________ Exp. date ________
___Bill my BAC No. ____________ Exp. date ________

Total for all books checked $ ________
Postage, 25 cents per book for ________ books $ ________
Grand Total $ ________

Name ____________
Address ____________
City ____________ State ______ Zip ______
Signature ____________

You may photocopy this page if you wish to leave your BYTE intact. Please allow six weeks for delivery.

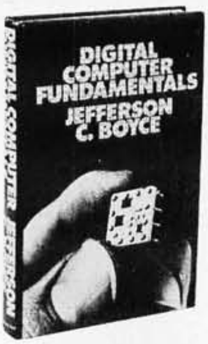
Circle 4 on inquiry card.



Adam Osborne's books **An Introduction to Microcomputers, Volumes 1 and 2,** are a concise compendium of the technical details of microprocessors at the component (engineering) level. These are the source books for the system designer who plans to employ the microprocessor, or the advanced homebrewer who wants a dash of customization not found in commercial products.

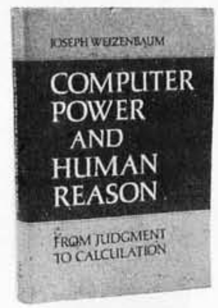
____ Volume 1 is subtitled "Basic Concepts." This is the book which presents a framework of ideas concerning the design and use of small computers implemented with LSI. Topics include definitions of the microcomputer, fundamental concepts of logic and numbering characteristics of instruction sets, etc. $7.50.

____ Volume 2 is a much thicker (895 page) detailed volume which complements the information in the first volume. This is the volume which fills in many of the details left out of the conceptual treatment in Volume 1. Here you'll find 19 detailed chapters on the engineering and logical specifications of products made by 16 different manufacturers, including in many cases reprints from the manufacturers' documentation as well as new materials provided by the author. Published in 1976, it even includes such processors as the MicroNOVA by Data General and the Texas Instruments TMS-9900 as well as the older 8 and 16 bit machines. Organization is by design type, and where parts of several manufacturers were intended for a given processor design such as the 8080, these are grouped into a single chapter. $12.50



____ **Digital Computer Fundamentals** by Jefferson C Boyce. The way to a world of learning is through books. A great place to start, and to return from time to time, is the classical textbook. This new book from Prentice Hall is intended as just that. Topics covered include digital computer operation, basic computer circuits and concepts, Boolean algebra, implementing computer operations in hardware, communicating with the computer and related issues of coding schemes, detailed discussions of the control section, memory functions, arithmetic and logic functions, input and output functions of a classical computer, a chapter on computer programming and a final summary chapter on the details of a typical minicomputer design interpreted in the light of the more theoretical general concepts in the book. This book is excellent background information for the literate and well read hacker. Order yours today. $15.95 hardbound.

____ **How to Buy and Use Minicomputers and Microcomputers** by William Barden. People have often asked us where to turn to get an introductory book about computers for personal use. One excellent place to start is **How to Buy and Use Minicomputers and Microcomputers,** William Barden Jr's instant summary of the small computer revolution, published by Howard Sams in mid-1976. This is one of the first books of the "general introduction to computers" genre to be published with an emphasis towards the small computer and personal computing as it is being practiced these days. the book, written for the novice as well as the expert, surveys the technical details of the field in nine chapters and 10 appendices. This book is light (but essential) reading for the experienced computer person, and worthy of serious, concentrated perusal by the novice. $9.95.



____ **Computer Power And Human Reason** by Joseph Weizenbaum. This book is one which should be purchased or read for several reasons. If you're presently a programmer by trade or skill, you'll see a philosophy of computer use and abuse propounded. It's genuinely interesting, and definitely provocative if you reference the storm of letters, counter letters and counter counter letters which this book produced in the Association for Computing Machinery's *SIGART* newsletters during 1976. If you're a novice to the field, the tutorial and explanatory chapters of this book, which are aimed at the layman, will serve as an excellent background source which is also eminently readable. This includes an excellent and low level explanation of what an algorithm is, and how computers go about executing effective algorithms. $5.95 softbound.

Send to:

BITS, Inc
70 Main St
Peterborough NH 03458

Total for all books checked $ ________
Postage, 25 cents per book for ________ books $ ________
Grand Total $ ________

Check payment method:
____ My check is enclosed
____ Bill my MC No. ________________ Exp. date ________
____ Bill my BAC No. ________________ Exp. date ________

Name ________________
Address ________________
City ________________ State ______ Zip ______
Signature ________________

You may photocopy this page if you wish to leave your BYTE intact. Please allow six weeks for delivery.

Circle 4 on inquiry card.

Continued from page 52

Selectric mechanism found in the Keyboard Printer is a set of switch contacts which are closed by movement of the tilt rotate bails, and by movement of the cams in various stages of the printing cycle. These contacts can also be seen in photo 4. Again, no electric power is applied to these contacts inside the Selectric, but six of them, called C1 to C6, are wired together thru certain pins in the receptable at the back of the machine (more on this later). For printed output, these contacts can be tested to determine when the printing cycle is complete. For keyboard input, there is another set of contacts which must be tested at the proper instant in order to capture the code for the key just depressed. Other contacts are provided which make it possible to determine whether the machine is currently locked in upper or lower case, whether the end of line margin stop has been reached, and so on. According to the documentation, the contacts are rated for 40 mA at 10 V (minimum) to 300 mA at 48 V (maximum).

BCD and Correspondence Machines

At this point, I should clear up the mystery surrounding the differences between the so-called "BCD" and "Correspondence" versions of the Selectric Keyboard Printer. There are differences in three areas:

1. The arrangement of characters on the typeball that is used.
2. The arrangement of the fingers on the interposers connected to particular keys.
3. The code obtained for keyboard input at the 50 pin receptacle when a key is pressed.

The Correspondence version is the simpler of the two. All of the office typewriters are built this way, and nearly all the typeballs available from IBM use the Correspondence arrangement of characters. In a Correspondence encoded Keyboard Printer, the tilt and rotate bail contacts are wired directly to the 50 pin receptacle, and so the code obtained when a key is pressed is the actual tilt rotate code. Note that the tilt rotate code is the same for, say, an upper case A and a lower case a, so the current state of the shift contacts must be checked whenever a character is read.

Many Selectric Keyboard Printers were built for use in equipment which employed a 6 bit byte and the old BCD (binary-coded decimal) character code, and so IBM developed the "BCD" version of the Selectric. In this machine, the tilt and rotate contacts (there are several sets of contacts for each bail) are wired through a maze of diodes and shift contact connections to yield a unique 6 bit code for all of the essential characters in the BCD set. Hence the code which reaches the 50 pin receptacle can be read directly into a 6 bit byte, and the shift contacts themselves need not be tested. Of course, a 6 bit byte can represent only 64 different characters, and after allowing for the digits and various special characters, there was room for only the upper case alphabetics. In fact, because of the limitations of wiring through diodes and switch contacts, only 48 distinct codes are actually produced. Even so, in order to accomplish this wiring feat, it was necessary to move some of the essential characters to convenient spots on the typeball, and hence the interposers with certain finger combinations also had to be moved around in order to preserve the usual layout of the keyboard. This is why the characters are all mixed up when you type manually on a BCD machine with a Correspondence typeball. Indeed, just to make everything fit together, IBM puts only the upper case characters on most of the typeballs intended for use with the BCD machine. (An exception is the Model 963 typeball which is used in many timesharing terminals.) But, in fact, the mechanism is still capable of tilting and rotating to any character position.

What does all this mean for the computer hobbyist? If you are using the Selectric as a printer only, it makes no difference whether you have a BCD or a Correspondence machine, since in either case you have direct access to the tilt and rotate magnets. By energizing the proper combinations of the seven magnets, you can use both BCD and Correspondence typeballs with either machine. (My Selectric is a BCD machine and I regularly use it with a Correspondence encoded Courier 72 typeball.)

If you want to use the Selectric keyboard for computer input (and you want upper and lower case), or if you want to use the machine off line with a variety of Correspondence encoded typeballs, you are considerably better off with the Correspondence version of the Keyboard Printer. But, since most of the units available through surplus channels (at least at reasonable prices) are BCD machines, you may have to settle for one of these. With some mechanical and electronic skill (and lots of courage), you could convert a BCD machine into a Correspondence version by:

1. rearranging the interposers to match the Correspondence typeball arrangement.

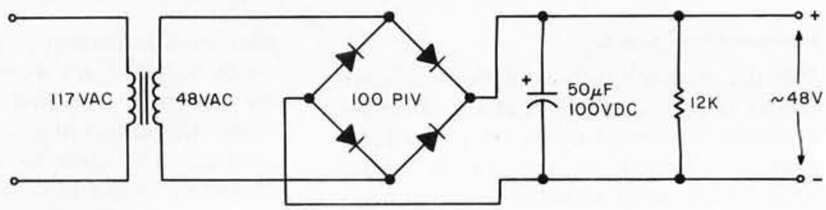


Figure 2: A very simple power source for the unregulated DC used to power the solenoids of the Selectric Keyboard Printer.

2. tearing out all the wiring for BCD code generation and replacing it with direct connections from the bail switch contacts to the 50 pin receptacle.

So much for the theory of operation of the Selectric mechanism. Now let's get on to the design of an interface unit which will let us control the Selectric printer using standard TTL level signals from a computer output port. Mindful always of our potential exposure to Murphy's Law, we will keep this interface as simpleminded as possible. Readers with more sophistication in electronics may use this approach as a jumping-off point (so to speak) for their own designs.

Interface Design

To control the operation of the Selectric printer we must provide three types of functions:

1. Signal conversion of TTL levels to magnet currents.
2. Code conversion of ASCII codes to tilt rotate code.
3. Control and timing to type successive characters, wait for carriage return, etc.

It seemed to me that the most appropriate division of labor was to provide the first function in hardware, and the second one in software. Signal conversion requires an external power source, while code conversion requires some flexibility to accommodate different typeballs. For the third function, I have experimented with both open loop control (realized entirely in software) and closed loop control (which uses a hardware feedback signal); both approaches will be discussed briefly here.

Signal Conversion

For signal conversion, we simply need a power source for the Selectric magnets and a means of switching the power on and off using TTL level signals. For the power source, we need a maximum of about 1 A of DC (for seven simultaneously energized magnets at 125 mA per magnet) in the range of 43 to 53 V. The source need not be regulated nor even filtered. (See "Watts Inside a Power Supply," by Gary Liming, January 1977 BYTE, page 42, for a further discussion.) Figure 2 is a circuit diagram for the power supply which I built around a $4 surplus transformer. The only really essential element is the full wave rectifier. The capacitor was included simply to jack up the voltage of the particular transformer I was using to the point where it would energize the magnets.

To switch power on and off, I used a set of reed relays (optoisolators or power transistors could be used instead). These particular reed relays have a coil resistance of 290 ohms, so they can be driven by an ordinary TTL gate (17 mA at 4.8 V, or 10 TTL loads). They are available from Digi-Key Corporation, POB 677, Thief River Falls MN 56701, for $1.70 each (part number 5VRR). I used a total of 12 relays, six for the print magnets (since I forgot about the "check" magnet) and six for the most important control functions (space, backspace, tab, carriage return, and upper and lower case shift).

The reed relays were each connected to a computer output port and a Selectric magnet through the circuit diagram shown in

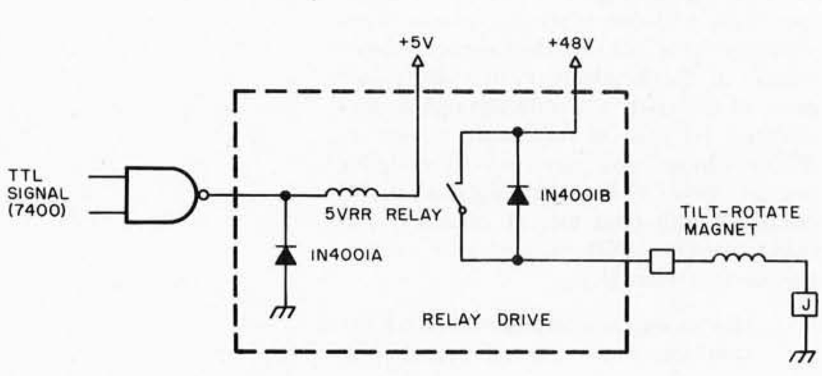


Figure 3: Switching of the solenoid actuator magnets in the Selectric Keyboard Printer is accomplished by this basic circuit. A reed relay which is within the drive capabilities of TTL is driven from a TTL logic gate, with protection against back EMF provided by the diode A. The reed relay, in turn, drives the magnet in the printer from the 48 V (nominal) supply of figure 2. Diode B provides back EMF protection for the relay contacts to prevent arcing which would shorten the life of the relay. The dotted line outlines the detailed circuit repeated many times in figure 4.

figure 3. Here the 1N4001 diodes protect the TTL gate and the reed switch from voltage transients in the two coils. Since I needed a standard TTL buffer to provide enough current for each reed relay, and since I wanted to economize on my use of output ports, I used a seventh control line to switch between the six print magnets and the six control function magnets. The resulting circuit diagram is shown in figure 4. The lettered squares which terminate the reed switch contact lines refer to pin designations on the Selectric's 50 pin receptacle (see below). Photo 5 shows the physical layout of the components of figure 4 in the interface which I built. Most of the wiring is Vector Slit n' Wrapped on the other side of the square piece of Vectorboard.

This construction layout is not recommended! Allow yourself much more room for repairing, replacing or adding components (like a seventh pair of reed relays!). A length of scrapped telephone cable makes a good connection between the interface and the Selectric itself. Also shown in photo 5 is a 50 pin connector which plugs into the

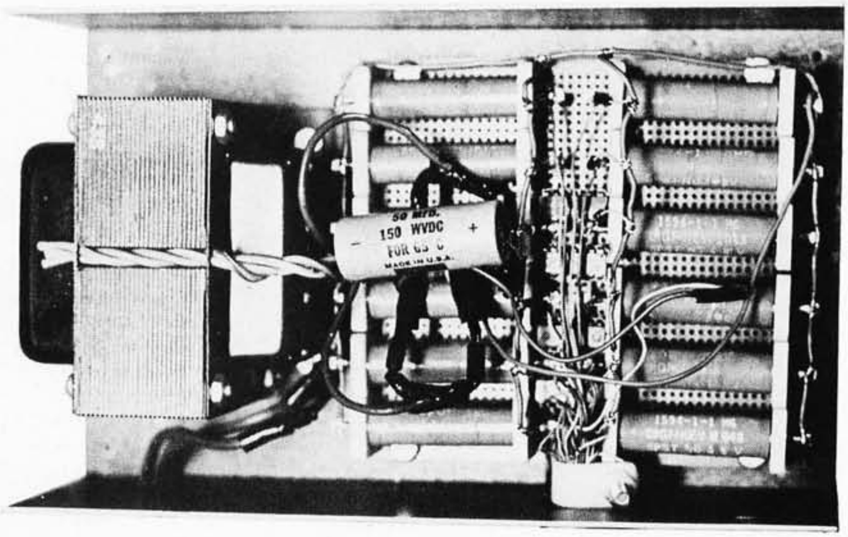


Photo 5: Physical layout of the components of the interface box which houses the circuit described in this article.

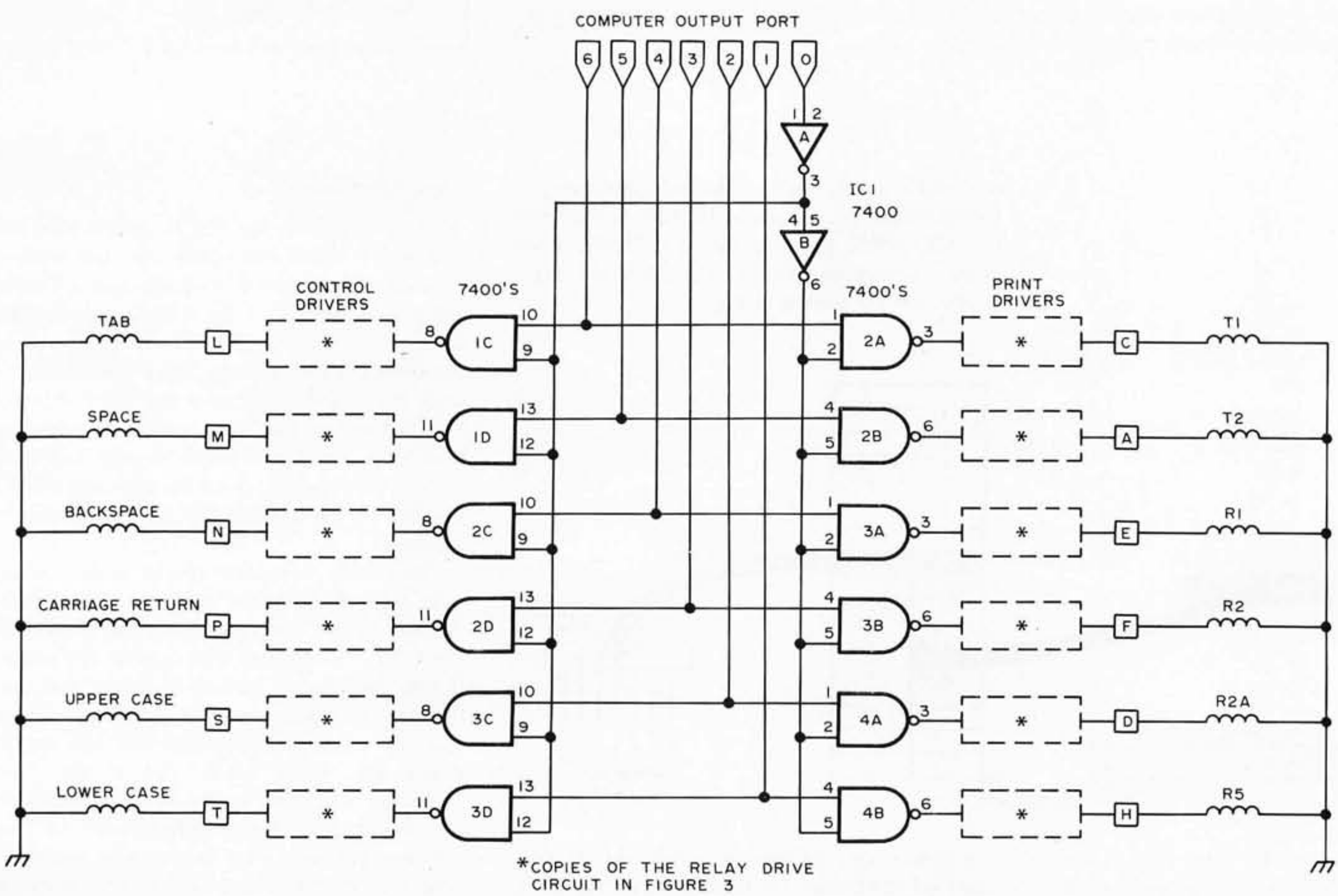


Figure 4: The complete interface schematic. The 7400 NAND gate logic is used to select either the drivers for the miscellaneous control functions, or the drivers for the print commands. The basic drive circuit of figure 3 is repeated once for each magnet in the printer.

Pin		Function
A	←	T2
B	←	Check
C	←	T1
D	←	R2A
E	←	R1
F	←	R2
H	←	R5
J	←	Magnet Common
K	←	Keyboard Lock
L	←	Tab
M	←	Space
N	←	Backspace
P	←	Carriage Return
R	←	Index
S	←	Upper Case Shift
T	←	Lower Case Shift
U	←	Red Ribbon Shift
V	←	Black Ribbon Shift
W	→	C1 N/C
X	→	Contact Common
a	→	Feedback N/C
b	→	Feedback N/O
e	→	End of Line N/C
f	→	End of Line N/O
n	→	C1 N/O
r,s,t,u,v,w	→	BCD Bit Lines

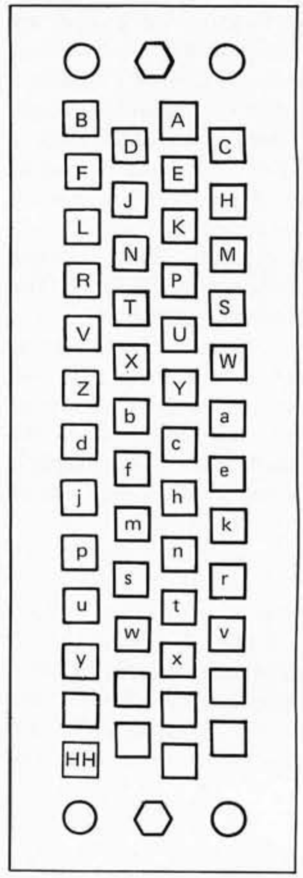


Figure 5: The Selectric Keyboard Printer receptacle pin identifications. This receptacle can be purchased as a spare part through an IBM office. The arrows in this table indicate direction of the signal: A left arrow indicates drive to the printer (typically a magnet) from a source in the interface; a right arrow indicates a sensor contact in the printer.

receptacle at the back of the Selectric, which I obtained from my local IBM branch office for $20 (IBM part number 1167134). The more important pin designations on this connector are shown in figure 5.

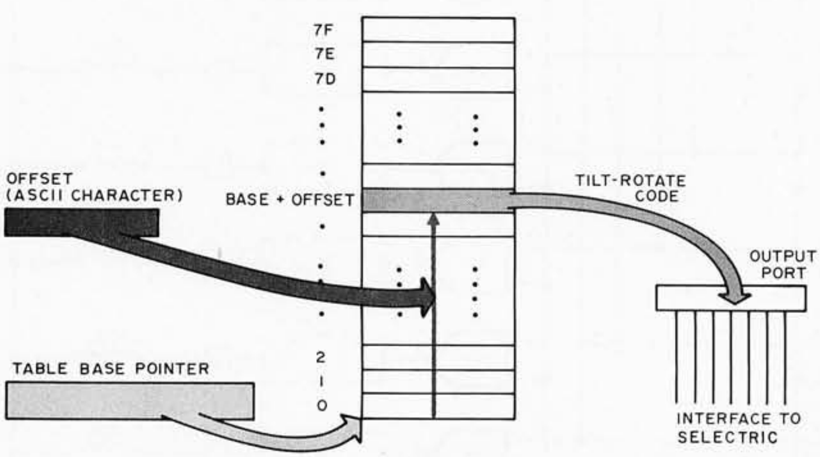


Figure 6: Table structure for the conversion of ASCII to Selectric coding. The table base pointer identifies the start of the table. There should be one table for each different ball coding scheme employed. The ASCII character value is added to the base address giving an address in the table. At this address is found the code which is sent to the output port. The logic of sending the code to the output port is given in detail by figure 8.

Code Conversion

Assuming that the ASCII code is used for characters inside the computer, the process of code conversion is basically just a simple table lookup: The 7 bit ASCII code is used as an index into a 128 byte table to obtain the 6 bit tilt rotate code. Since the tilt rotate code for a given character may vary depending on the typeball that is used, it should be possible to switch between several 128 byte tables. This is easily done by indexing from a pointer to the base of the table as shown in figure 6.

The main complication in code conversion is the handling of upper and lower case. At any given time the Selectric Keyboard Printer is locked into one case or the other. If the machine is locked in upper case and the next character to be printed is an upper case A, we need only send out the appropriate tilt rotate code. But if the next character is a lower case a, we must energize the lower case shift magnet, wait for the machine to shift into lower case, and then send out the tilt rotate code. This is easily accomplished by using a seventh bit in the table entry byte for each ASCII character to indicate whether it is to be printed in upper or in lower case.

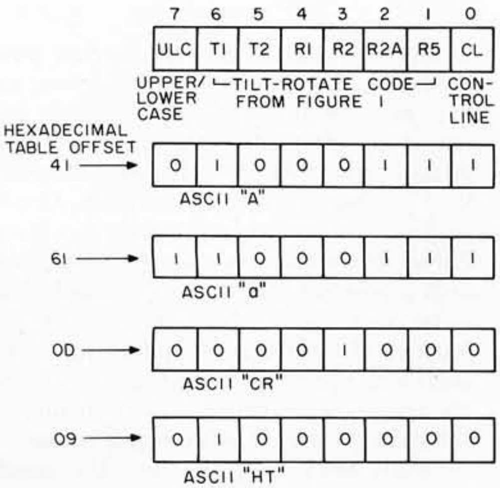


Figure 7: The coding scheme for each conversion table entry is given by the general box at the top of this diagram. Bit 7 tells the software whether the mechanism should be in the upper or lower case mode. (The need to shift explicitly in a Selectric is reminiscent of the shift requirements of Baudot Teletypes.) The tilt rotate code contained in bits 6 thru 2 is derived from figure 1 for each character in the table. (For other ball arrangements, a version of figure 1 would need to be generated.) The low order bit of the word is used to indicate to the logic of figure 4 whether a control command (0) or print command (1) is being sent.

The last problem in code conversion is the handling of control functions such as carriage return, tab, backspace, etc. Fortunately, the ASCII character set assigns unique 7 bit codes for functions such as these. For example, the ASCII carriage return character (hexadecimal code 0D) can be used for carriage return, and the ASCII horizontal tab (hexadecimal code 09) can be used for the tab function. Since in my interface a special control line determines whether the six output ports affect the print magnets or the control function magnets, I can use the eighth bit in each table entry byte to set the control line appropriately. The table entries for the printable characters have this bit set to 1, with six bits providing the tilt rotate code; the entries for the control characters have this bit set to 0, with the bit corresponding to the given control function magnet set to 1 and the other five bits set to 0. This encoding is illustrated in figure 7.

Once we have this encoding of the information needed for code conversion, the actual program logic to accomplish the conversion is straightforward. A flowchart of the logic is presented in figure 8, and an

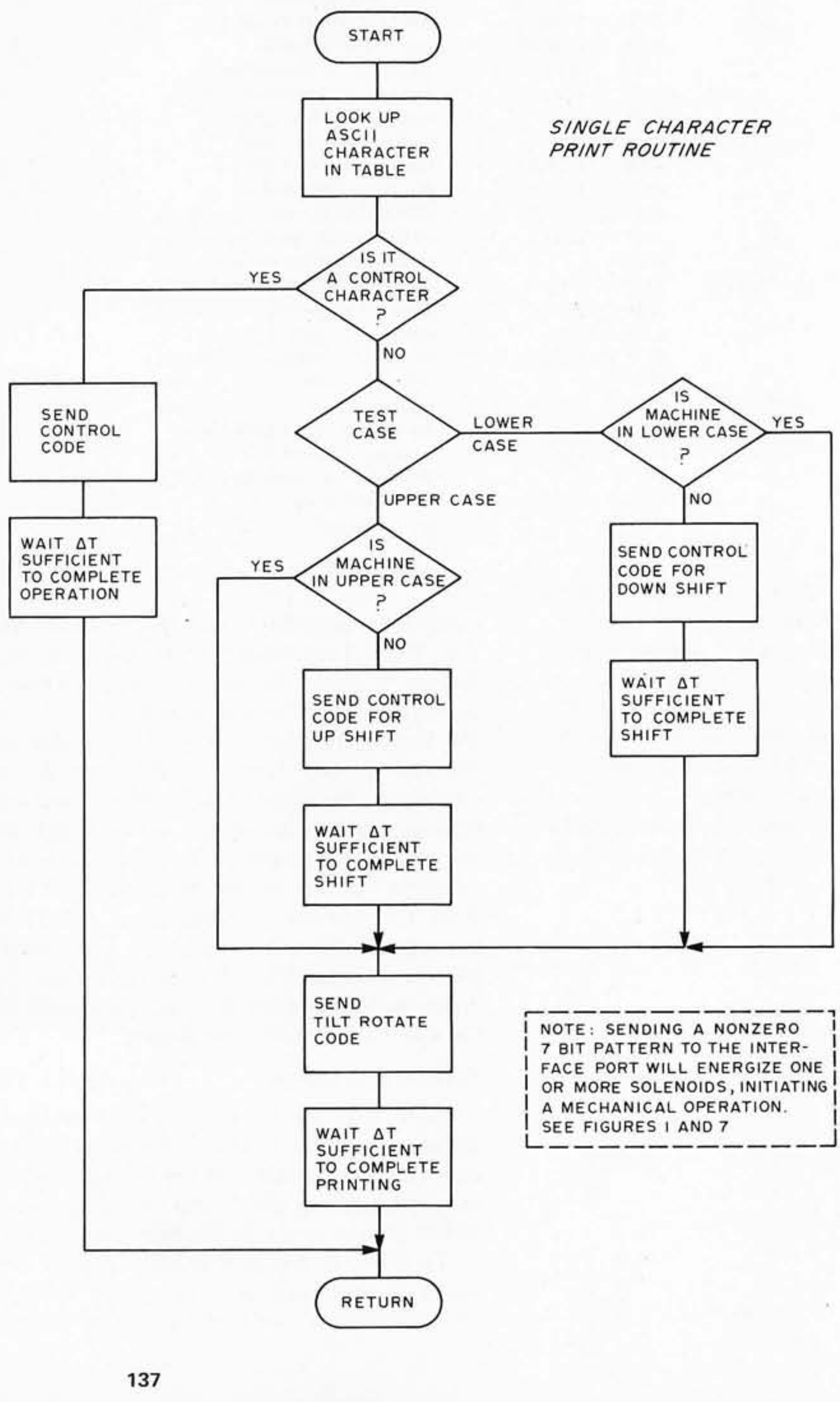


Figure 8: A flowchart giving the logic of a simple open loop driver program which takes a given ASCII character, looks up its table entry, and then takes appropriate printer actions. As an open loop program, each time delay in this chart (the ΔTs) is picked to reflect the worst case response time for the action involved. This makes the Selectric type successfully, but does not optimize operation for the maximum speed, since as everyone knows, the worst case is often not identical with the typical value of a parameter.

CHARACTER OUTPUT ROUTINE FOR SELECTRIC KEYBOARD PRINTER

```
OUTCH   TAY                  ASCII character to index register
        LDA   (TABPT),Y      get code byte from table
        LSR   A              test low order bit
        BCC   CTL            0 means control character
        ROL   A              test high order bit
        BMI   LOWER          1 means lower case character
        LDX   #4             code for upper case shift
        LDY   CASE           check current case
        BEQ   OK             0 means upper case
        INC   CASE           indicate shift to upper case
        JMP   SHIFT          go initiate shift operation
LOWER   LDX   #2             code for lower case shift
        LDY   CASE           check current case
        BNE   OK             -1 means lower case
        DEC   CASE           indicate shift to lower case
SHIFT   STX   PORT           send shift code to port
        JSR   ENERG          for 10 milliseconds
        LDY   #60            delay for 60 milliseconds
        JSR   WAIT           until shift operation is done
OK      STA   PORT           send tilt rotate to port
        JSR   ENERG          for 10 milliseconds
        LDY   #50            delay for 50 milliseconds
        JSR   WAIT           until print operation is done
        RTS                  return to calling program
CTL     ROL   A              restore control code
        STA   PORT           send to output port
        JSR   ENERG          for 10 milliseconds
        LDY   #120           delay for 120 milliseconds
        JSR   WAIT           until control operation is done
        RTS                  return to calling program
ENERG   LDY   #10            set up for 10 millisecond delay
        JSR   WAIT           loop for that long
        LDY   #0             send 0s to output port
        STY   PORT           to turn off magnet current
        RTS                  return to caller
WAIT    LDX   #200           number times thru inner loop
LOOP    DEX                  decrement inner loop count
        BNE   LOOP           loop until count is 0
        DEY                  decrement outer loop count
        BNE   WAIT           loop until count is 0
        RTS                  return to caller
```

Listing 1: 6502 assembly language source code of a program which implements the logic of the flowchart in figure 8. This program is a subroutine which will drive the Selectric Keyboard Interface in an open loop mode and is run on a KIM-1 system.

equivalent assembly language program for the MOS Technology 6502 used in my system is shown in listing 1. In this simple version of the program, delay loops are used for timing purposes, and sufficient time is allowed either to print a character or to complete the worst case control function (carriage return across the entire length of the page). Of course, this version of the program will operate the Selectric at far less than its maximum rated speed, and will monopolize the processor's time while waiting for completion of each operation. In order to improve on this, we turn next to the subject of control and timing.

Control and Timing

Now that we have a working Selectric interface, we can turn our attention to two major improvements: driving the Selectric at maximum rated speed, and minimizing use of the processor's time for Selectric control.

To drive the Selectric at full speed we can adopt an approach of "open loop" control or "closed loop" control. Open loop control involves keeping track of the carriage position, margin, tab stops and similar information in software (changing the margin and tab stop information via software interpreted commands), and calculating the delay time necessary for each operation. Closed loop control involves testing the Keyboard Printer's switch contacts to determine when each operation has been completed. The worst case delay approach used in the program of listing 1 is a simplified version of open loop control. For full speed operation, the closed loop approach is much simpler and more reliable; so let's consider it here.

Nearly every mechanical operation opens or closes some set of switch contacts inside the Selectric. Sets of contacts are wired to the 50 pin receptacle in a variety of ways to reflect operations such as printing, tabbing, backspacing, etc. We will not consider all the possible methods of achieving feedback control using these contacts, but will outline one particularly simple approach, which remains to be tested in my own system. The pin labeled a on the receptacle is wired through a set of normally closed contacts, and the pin b through corresponding normally open contacts, associated with the set of common contacts connected to pin X. Figure 9 shows how these contacts may be debounced to yield a clean TTL level signal (ignoring the nominal voltage ratings for the contacts). Here we use the last half of the 7400 package left over from figure 4. During any printing or control function operation, pin a will go from ground to +5 V and back to ground again, while pin b does the reverse. Hence the feedback line will go from logic 1 to 0 to 1. By sensing this change in software through a loop testing the feedback input port after energizing the magnets, we can closely control the operation. When the line goes to logic 0, we can turn off current to the magnets, and when it returns to logic 1, we are ready to start the next operation.

The second problem we face in control and timing is how to minimize use of the processor's time for Selectric control. Here, of course, is where the interrupt system comes into play. If we are using the circuit outlined in figure 9 for closed loop control, we can tie the feedback line to a processor interrupt rather than to a data input port. If we are relying instead on open loop control, we can use a programmable interval timer which is capable of causing an interrupt as an alternative to delay loops. The software to handle interrupts from the Selectric is slightly complicated by the need to shift between upper and lower case prior to typing the next character, but this can be handled by initiating the shift operation and then arranging to retry the character printing operation on the next interrupt, at which time the Selectric will be locked into the proper case.

Actual Experience

Hopefully this article has given the reader all the information he or she needs to build a Selectric Keyboard Printer interface similar to, or better than mine. Lest you are unduly emboldened by the foregoing discussion, however, consider what can go wrong.

I carefully tested the interface in stages, by using an ohmmeter to verify that bit patterns sent to my computer output port closed the proper combinations of reed switches, and by testing the power supply on some of the Selectric's magnet coil connections. Nevertheless, when I first tested the entire setup, I thought I saw a blue flash around one of the reed relays when I tried to pulse the R2 magnet. Nothing seemed to happen when I tried again, except that the R2 magnet wasn't being energized. Then, listening carefully, I heard a telltale simmering sound that sent me leaping for the electric outlet. The R2 reed relay had stuck closed, and on further examination I found that most of the arc suppressing diodes inside the Selectric had been destroyed. After painstakingly replacing the R2 reed relay and installing the diodes visible in photo 5, I tried again. This time I found out why the reed relay, like its replacement, was sticking closed! The R2 magnet in the unit I purchased had been burned out and was a short circuit. No wonder the unit was a surplus item.

Not willing to give up, I managed to remove the coil from the R2 magnet core, and replace it with the coil from the unused (by me!) check magnet. After this feat, I found that when I typed manually on the keyboard, only @s, 0s, and a few other characters could be printed! Only after hours of reading and experimentation did I discover that the adjustment of the plate holding the magnet armatures in place (which I had removed to change the coils) was critical, and could be set only by considerable trial and error.

These are the kinds of things that can go wrong. You cannot be too careful in playing with these machines! Readers certainly should investigate the possibility of an IBM maintenance contract on at least the mechanical portion of the Keyboard Printer, which need not be too expensive.

And, to conclude, although I probably never would have undertaken this project had I known at the outset what it would ultimately entail, it certainly is satisfying to have that Selectric typing away under the control of my home computer. To anyone else who is ready to undertake such a project, I hope that this article has helped, and I wish you the best of luck.■

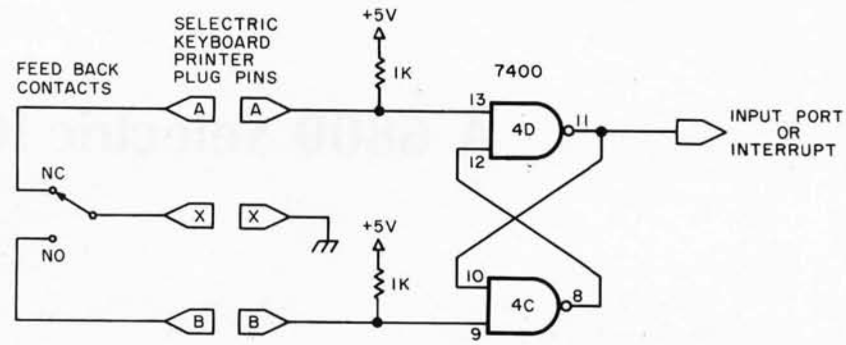


Figure 9: A circuit for debouncing the feedback information generated by contacts in the printer which are mechanically linked to the action. Using the feedback pulse to drive an input port or interrupt line can result in operation at the maximum possible speed since the timing is now on an "each case" basis rather than "worst case."

BIBLIOGRAPHY

"IBM Selectric Input-Output Writer: An Exciting Advance in the Field of Input-Output Media," Form # 543-0033-1. This manual is absolutely essential since it gives circuit diagrams, timing charts, and end views of the magnets and switch contacts.

"IBM Selectric I/O Keyboard Printer: Customer Engineering Manual of Instruction," Form #241-5159-3. This or a similar manual is very valuable for understanding the mechanical functioning of the Keyboard Printer.

A 6800 Selectric IO Printer Program

Listing 1: The listing of the Selectric printer interface routine for a 6800 system driving the IO version of the standard office typewriter. This listing is extracted from two assemblies done using the Southwest Technical Products Corporation's version of the M6800 self assembler. The first part of the listing is the actual code, and the second part is a table of Selectric correspondence codes which is referenced using ASCII codes as an index into the table which is computed at CONV1.

```
SWTPC M-6800 ASSEMBLER
ENTER PASS : 1P,1S,2P,2L,2T

                          NAM    SELECTRIC
                   *SELECTRIC DRIVER PROGRAM FOR SWTPC 6800 ASSEMBLER
                          OPT    S
                          OPT    O
                          OPT    L
0100                      ORG    $0100
0100 7E 17CD              JMP    START
0212                      ORG    $0212
0212 BD 17F4              JSR    START1     CALL OUTPUT(NEW)
08F7                      ORG    $08F7
08F7 18FF                 FDB    $18FF      MAKE ROOM FOR PATCH
093D                      ORG    $093D
093D 1900                 FDB    $1900
13D6                      ORG    $13D6
13D6 1900                 FDB    $1900
17CD                      ORG    $17CD
17CD C6 FF       START    LDA B  #$FF       INITIALIZE PIA
17CF F7 8000              STA B  PIAOUT
17D2 7F 8002              CLR    PIACHK
17D5 C6 04                LDA B  #$04
17D7 F7 8001              STA B  PIAOUT+1
17DA F7 8003              STA B  PIACHK+1
17DD C6 81                LDA B  #$81       START ALWAYS IN LOWER CASE
17DF F7 8000              STA B  PIAOUT
17E2 F7 18CA              STA B  CASE
17E5 FE 181A              LDX    COUNT1     SETUP TIMER FOR SHIFT CYCLE
17E8 FF 18CB              STX    COUNTR
17EB BD 18C3              JSR    TIMER
17EE 7F 8000              CLR    PIAOUT
17F1 7E 0300              JMP    $300       GOTO MAIN PRGM
17F4 84 7F       START1   AND A  #$7F       RESET PARITY
17F6 FF 18CD              STX    SAVEX      SAVE XREG FOR MAIN PRGM
17F9 81 10                CMP A  #$10       TRAP HOME-UP
17FB 26 02                BNE    CR
17FD 20 04                BRA    CR1        PRINT IT AS CR,LF
17FF 81 0D       CR       CMP A  #$0D       TRAP CR
1801 26 0B                BNE    SP
1803 86 84       CR1      LDA A  #$84
1805 FE 181C              LDX    COUNT2     SETUP TIMER FOR CR,LF
1808 FF 18CB              STX    COUNTR
180B 7E 18BA              JMP    EX1
180E 81 20       SP       CMP A  #$20       TRAP SPACE
1810 26 05                BNE    CONVO
1812 86 88                LDA A  #$88
1814 7E 18B4              JMP    EX2        GO PRINT BUT DO NOT RESET MSB
1817 7E 1880     CONVO    JMP    CONVRT
181A 2000        COUNT1   FDB    $2000
181C 1000        COUNT2   FDB    $1000
181E 0400        COUNT3   FDB    $0400
1880                      ORG    $1880
1880 81 20       CONVRT   CMP A  #$20       IS IT A PRINTING CHARACTER?
1882 22 02                BHI    CONV1      YES
1884 20 39                BRA    EXIT       NO
1886 B7 18D0     CONV1    STA A  TABLEP+1   CONVERT CODE
```

The following letter and listing 1 were received from an Italian reader of BYTE, Fulvio Guzzon of Rome. Fulvio purchased the same print mechanism (IBM Model 735 IO typewriter) which is described by Dan Fylstra in his article in this issue. We're treating Fulvio's letter as a short article, since its technical content is far above that of the usual letter. The listings photographically reproduced here were typed on pin feed paper using his printer mechanism. The text of his letter was submitted using a text editor with the Selectric IO mechanism as its output.

I understand there is some interest among your readers in using a Selectric typewriter for hard copy. As you can see I have funneled an editor program (SWTPC) and an assembler program (SWTPC, too) through a Selectric typewriter. *[The original of this note was typed on the Selectric.]* I bought the machine on the surplus market in Boston and it had some problems: It was stuck in upper case by a bolt screwed on the right side of the frame, it had some unrecoverable backlash in the head rotate mechanism, and many feedback and interlock contacts were missing or badly damaged. I had the machine serviced here in Rome (Italy) and at last, with a new carriage, a new motor (here we have 220 V 50 Hz power), and a new set of shift magnets, the printer was ready. I decided to use it only as a printer in order to reduce the hardware and software effort to a minimum.

On the underside of the machine there are seven printing magnets. In table 1 I have paired them with the bits from 0 to 6. Seven transistors provide for the interface between the PIA and the printer.

There are seven more magnets for the machine commands: space, backspace, tab, carriage return, index (line feed), upper case, and lower case; so seven more transistors are required. Seven output lines from the PIA in slot 0 are switched between the two sets of magnets by digital logic. The various feedback and interlock contacts were wired in series and filtered for bounce by a condenser and a software loop. The conversion table shown in the assembly listing provides for the characters used on the so called "Correspondence" balls. As I later found out, there are minor variations between the balls of this series.

The MSB in the table is set when the character to be printed is on the upper case half of the ball. (The upper or lower case of ASCII code bears no relation to the upper or lower side of Selectric golf balls). The MSB of the output byte to the printer

Listing 1, continued:

```
1889 FE 18CF          LDX    TABLEP
188C A6 00            LDA A  0,X
188E 26 02            BNE    CASECK   IS IT AVAILABLE SOMEWHERE ON
                * THE BALL?
1890 20 2D            BRA    EXIT     NO,RETURN
1892 2A 04     CASECK BPL    CASELW   MSB CLEAR?
1894 C6 C0            LDA B  #$C0     NO,CHECK IF PRINTER IS IN UC
1896 20 02            BRA    SKIP
1898 C6 81     CASELW LDA B  #$81     YES,CHECK IF PRINTER IS IN LC
189A F1 18CA   SKIP   CMP B  CASE     NEW CHAR. SAME HALFBALL
                *AS THE PREVIOUS ONE?
189D 27 13            BEQ    PRINT1   YES GO AND PRINT IT
189F F7 8000          STA B  PIAOUT   NO, ROTATE BALL 180 DEGREES
18A2 F7 18CA          STA B  CASE     AND RECORD IT
18A5 FE 181A          LDX    COUNT1   SETUP TIMER FOR SHIFT CYCLE
18A8 FF 18CB          STX    COUNTR
18AB 8D 16            BSR    TIMER
18AD 7F 8000          CLR    PIAOUT
18B0 8D 11            BSR    TIMER
18B2 84 7F     PRINT1 AND A  #$7F     RESET CASE BIT AND
18B4 FE 181E   EX2    LDX    COUNT3   SETUP TIMER FOR PRINT CYCLE
18B7 FF 18CB          STX    COUNTR
18BA B7 8000   EX1    STA A  PIAOUT   NOW PRINT
18BD 8D 12            BSR    WAIT1
18BF FE 18CD   EXIT   LDX    SAVEX    RESTORE X REG
18C2 39               RTS             GO AND FETCH NEXT CHARACTER
18C3 FE 18CB   TIMER  LDX    COUNTR
18C6 09        LOOP   DEX
18C7 26 FD            BNE    LOOP
18C9 39               RTS
     8000      PIAOUT EQU    $8000
     8002      PIACHK EQU    $8002
18CA 00        CASE   FCB
18CB 0000      COUNTR FDB
18CD 0000      SAVEX  FDB
18CF 1800      TABLEP FDB    $1800
18D1 8D F0     WAIT1  BSR    TIMER
18D3 C6 01            LDA B  #1       SETUP MASK
18D5 F5 8002          BIT B  PIACHK   PRINT CYCLE STARTED?
18D8 27 F7            BEQ    WAIT1    NO
18DA 7F 8000          CLR    PIAOUT   YES ON IT'S WAY
18DD 8D E4     WAIT2  BSR    TIMER
18DF F5 8002          BIT B  PIACHK   READY FOR A NEW ONE?
18E2 26 F9            BNE    WAIT2    NO
18E4 39               RTS             YES!
18E8                  ORG    $18E8    SHIFT START OF SYMBOL TABLE
                *TO NEXT PAGE
18E8 C1               FCB    $C1
18E9 20               FCC    5,
18EA 20
18EB 20
18EC 20
18ED 20
18EE FFFF             FDB    $FFFF
18F0 C2               FCB    $C2
18F1 20               FCC    5,
18F2 20
18F3 20
18F4 20
18F5 20
18F6 FFFF             FDB    $FFFF
```

Listing 1, continued:

```
00108 18F8 D8              FCB    $D8
00109 18F9 20              FCC    5,
      18FA 20
      18FB 20
      18FC 20
      18FD 20
00110 18FE FFFF            FDB    $FFFF
00111                      END

START  17CD
START1 17F4
CR     17FF
CR1    1803
SP     180E
CONVO  1817
COUNT1 181A
COUNT2 181C
COUNT3 181E
CONVRT 1880
CONV1  1886
CASECK 1892
CASELW 1898
SKIP   189A
PRINT1 18B2
EX2    18B4
EX1    18BA
EXIT   18BF
TIMER  18C3
LOOP   18C6
PIAOUT 8000
PIACHK 8002
CASE   18CA
COUNTR 18CB
SAVEX  18CD
TABLEP 18CF
WAIT1  18D1
WAIT2  18DD

TOTAL ERRORS 00000

SWTPC M-6800 ASSEMBLER
ENTER PASS : 1P,1S,2P,2L,2T

00001                      NAM    TABLE
00002                      OPT    L
00003                      OPT    S
00004 1821                 ORG    $1821
00005 1821 FE              FCB    $FE
00006 1822 D4              FCB    $D4
00007 1823 BE              FCB    $BE
00008 1824 CE              FCB    $CE
00009 1825 D6              FCB    $D6
00010 1826 D8              FCB    $D8
00011 1827 54              FCB    $54
00012 1828 86              FCB    $86
00013 1829 C6              FCB    $C6
00014 182A 9E              FCB    $9E
00015 182B B0              FCB    $B0
00016 182C 18              FCB    $18
00017 182D 01              FCB    $01
00018 182E 34              FCB    $34
00019 182F 48              FCB    $48
00020 1830 46              FCB    $46
00021 1831 7E              FCB    $7E
00022 1832 36              FCB    $36
00023 1833 3E              FCB    $3E
00024 1834 4E              FCB    $4E
00025 1835 56              FCB    $56
00026 1836 16              FCB    $16
00027 1837 5E              FCB    $5E
00028 1838 1E              FCB    $1E
00029 1839 06              FCB    $06
00030 183A D8              FCB    $D8
00031 183B 58              FCB    $58
00032 183C 00              FCB    $00
00033 183D 30              FCB    $30
00034 183E 00              FCB    $00
00035 183F C8              FCB    $C8
00036 1840 B6              FCB    $B6
00037 1841 9C              FCB    $9C
00038 1842 82              FCB    $82
00039 1843 9A              FCB    $9A
00040 1844 DA              FCB    $DA
00041 1845 D2              FCB    $D2
00042 1846 B8              FCB    $B8
00043 1847 F8              FCB    $F8
00044 1848 C2              FCB    $C2
00045 1849 94              FCB    $94
00046 184A F0              FCB    $F0
00047 184B 92              FCB    $92
00048 184C CA              FCB    $CA
00049 184D FC              FCB    $FC
00050 184E B2              FCB    $B2
00051 184F CC              FCB    $CC
00052 1850 D0              FCB    $D0
00053 1851 90              FCB    $90
00054 1852 DC              FCB    $DC
00055 1853 C4              FCB    $C4
00056 1854 F2              FCB    $F2
00057 1855 BA              FCB    $BA
00058 1856 BC              FCB    $BC
00059 1857 84              FCB    $84
00060 1858 FA              FCB    $FA
00061 1859 C0              FCB    $C0
00062 185A F6              FCB    $F6
00063 185B 00              FCB    $00
00064 185C 00              FCB    $00
00065 185D 00              FCB    $00
00066 185E 00              FCB    $00
00067 185F 81              FCB    $81
00068 1860 00              FCB    $00
00069 1861 1C              FCB    $1C
00070 1862 02              FCB    $02
00071 1863 1A              FCB    $1A
00072 1864 5A              FCB    $5A
00073 1865 52              FCB    $52
00074 1866 38              FCB    $38
00075 1867 78              FCB    $78
00076 1868 42              FCB    $42
00077 1869 14              FCB    $14
00078 186A 70              FCB    $70
00079 186B 12              FCB    $12
00080 186C 4A              FCB    $4A
00081 186D 7C              FCB    $7C
00082 186E 32              FCB    $32
00083 186F 4C              FCB    $4C
00084 1870 50              FCB    $50
00085 1871 10              FCB    $10
00086 1872 5C              FCB    $5C
00087 1873 44              FCB    $44
00088 1874 72              FCB    $72
00089 1875 3A              FCB    $3A
00090 1876 3C              FCB    $3C
00091 1877 04              FCB    $04
00092 1878 7A              FCB    $7A
00093 1879 40              FCB    $40
00094 187A 76              FCB    $76
00095                      END

TOTAL ERRORS 00000
```

interface is set to select a machine command. Only one input line of the PIA is used to sample the status of the printer READY or BUSY.

Since the shift feedback and interlock contacts were missing, a timing loop provides for the timing here; however, for the carriage return it has been necessary to build an interlock contact to lock out the printing function till the completion of a carriage return which takes a variable time.

A commented assembler listing of the program driving the printer was written for a 6800 and assembled with output to my Selectric (see listing 1). It can be loaded after the original SWTPC tape has been read in. A refinement which could be added is to provide for motor on or off via software as the printer can be powered up only after the program is running. This is because the power up reset of the computer leaves the PIA LINES all programmed as inputs, ie: open circuited and this simultaneously turns on all the machine magnets.

Another refinement could be to sense via an unused input line if the motor is on or off and steer the output to a TV terminal when the printer is off. To probe into undocumented programs like the SWTPC assembler or editor, I used a little program which searches the memory for a particular string of bytes and prints out the address of the first byte when and if found. I think it can save lots of time.

Fulvio Guzzon
c/o L Alessio
Via Anassagora 63
Casalpalocco 00124
Rome ITALY■

Table 1: Assignment of bits.

BIT 6	ROTATE+1	when energized removes the ROTATE+1 latch
BIT 5	ROTATE+2	when energized removes the ROTATE+2 latch
BIT 4	ROTATE+2A	when energized removes the ROTATE+2 supplementary latch
BIT 3	ROTATE-5	when energized activates the ROTATE-5 latch
BIT 2	TILT 1	when energized removes the TILT 1 latch
BIT 1	TILT 2	when energized removes the TILT 2 latch
BIT 0	CHECK	this one unlatches the print clutch (and so does every one of the previous six)

What's New?

A Full Size Floppy Disk with Altair Interface

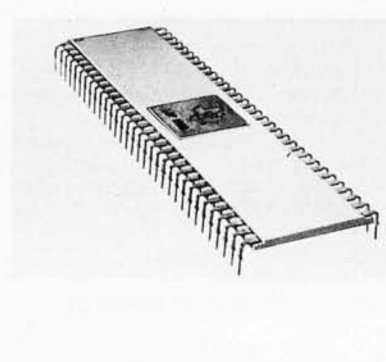
Peripheral Vision, POB 6267, Denver CO 80206, has announced this full size floppy disk for the Altair bus. Prices start at $750 for the interface card kit and one assembled and tested drive. A 24 V at 2 A power supply is also available in kit form for $45 or assembled for $65; and a cabinet is offered for an additional $85. The Peripheral Vision floppy disk interface card supports eight drives and according to the press release, stores over 300,000 bytes per floppy. A bootstrap EROM is included to make system start-up automatic.

The floppy is completely Altair bus compatible, and interface cabling is included. The Peripheral Vision floppy disk drive itself is manufactured by Innovex, and comes assembled and tested. A disk operating system with file management system is included on a floppy disk cartridge.■

Circle 615 on inquiry card.

New Technology for the 9900 Family

Texas Instruments Inc has introduced a new version of the 9900 architecture in the form of this SBP 9900, which uses integrated injection logic (called I^2L in much of the engineering and design literature). The key features of this new product are in the subtleties of using the



chip at a hardware level (software is identical to the previous TMS 9900 and Texas Instruments' 990 line of minicomputers which use this architecture). For the hardware designer or homebrew hacker, this new premium version of the design gives a wide (−55°C to +125°C) operational temperature range, infinitely variable clock rates from DC to 3 MHz (power consumption levels vary with the rate) and TTL compatibility with wide tolerances on power supply voltages. This is, however, a premium device intended for applications which require ruggedness. The ceramic packaged SBP 9900 starts at a $386 price in 100 piece quantities. Reader inquires should be directed to Texas Instruments Inc, Inquiry Answering Service, POB 5012, M/S 308 (attn: SBP 9900), Dallas TX 75222. ■

Circle 616 on inquiry card.

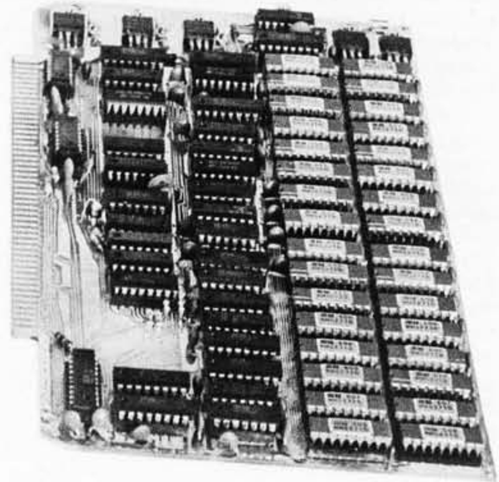
A Rugged Z-80 Product

Cromemco has sent along this press release picture of the new Z-2 processor which is the latest product of their laboratory. This processor uses their Z-80 processor board which according to the company is available in a fast 4 MHz version. Coupled with Cromemco's other peripherals which also work at this speed, in principle this one has one of the fastest processors yet available in a commercial product. The basic box contains the processor card, a mother board with 21 card slots for Altair compatible cards, and a heavy duty power supply intended to suffice for all system needs including floppy disk drives. Naturally, in addition to the $595 kit version of this processor, you'll want to have some dedicated monitor ROM and extra peripherals, but the price makes it an attractive way to start building a system. Cromemco has software of a monitor, assembler, and a BASIC with processor control extensions. The firm also makes numerous peripherals including digital to analog interface cards, ROM memory cards and a color graphic product. Cromemco is located at 2432 Charleston Rd, Mountain View CA 94043.■

Circle 617 on inquiry card.

A 16 K Byte Memory Board

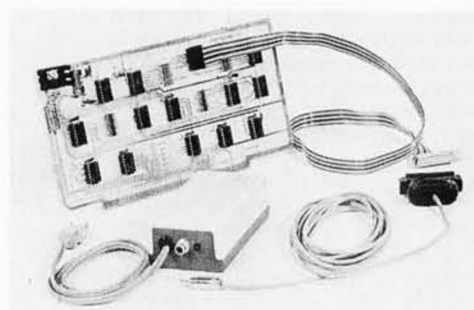
RHS Marketing, 2233 El Camino



Real, Palo Alto CA 94306, sends this picture of a new Altair compatible, assembled and tested 16 K byte memory board with a price of $485 or $30 per installed 1024 bytes. The board contains its own refresh control logic and uses 4 K dynamic memory chips with a total board power consumption of 5 W. External to the board, this product looks like a static memory board and has no wait states since the refresh is transparent. The number of personal computers with a saturated 64 K memory address space is likely to go up as more and more products such as this reach the market.■

Circle 618 on inquiry card.

Controlling Those Necessary Bells, Whistles and Other Goodies

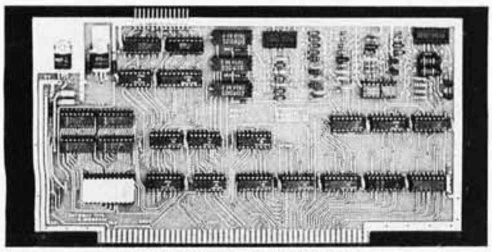


Comptek has designed and manufactured this interesting array of components with the needs of the real world interface in mind. The heart of the control setup is of course a typical microcomputer, an Altair bus compatible machine into which the main "control logic interface" board plugs. Out of the board comes a ribbon cable to a DB-25 connector at the back edge of the cabinet. The DB-25 is in turn the recipient of a cable which goes to your remote power unit, the box shown in this picture. This box, which is one of many which may be driven from one control logic interface board, contains the optically isolated 400 W controller for 110 VAC. The costs of this interface in kit form are $189 for a 16 channel

PC3216 control logic interface card, plus $39.50 for each PC3202 400 W power control unit kit. Other options include fully assembled versions at higher prices and a 32 channel PC3232 control logic interface card. A product like this is needed when you want to have your computer drive 110 VAC appliances, lighting circuits and other household electrical loads. Comptek is located at POB 516, La Canada CA 91011.■

Circle 619 on inquiry card.

Morrow Tape Interface



Morrow's Micro Stuff, POB 6194, Albany CA 94706, has introduced this Altair bus plug-in unit which generates and reads data on up to three channels of audio recording in the Kansas City standard of recording, at 300 bps. Also thrown into the board is a serial port to allow communication with a Teletype with reader control, as well as any RS-232 serial device. Also included is an 8 bit parallel board for use with parallel interfaced peripherals such as keyboards or tape readers. A ROM on the board holds 512 bytes of programming for the cassette interface, UART simulation, and transfer to or from your 8080's memory or the 512 byte programmable memory region on the board. This board is available in kit form for $120 or assembled and tested at $165 with warranty. The product is marketed by mail or through computer stores.■

Circle 620 on inquiry card.

Matrox Video Display with External Sync Capability

One fascinating possibility for the use of small computers is in combination with standard video signals for various purposes. (An example might be display of product feature data along with digital messages in a merchandising situation such as a department store.)

Matrox Electronic Systems, POB 56, Ahuntsic Sta, Montreal, Quebec CANADA H3L 3N5, has just announced the new member of its video RAM family of devices. This is the MTX-1632SL, shown here in front of a television driven by a mixed signal generated from another video source and the character generator. Since the horizontal and vertical synchronization can be slaved to the external source, such as the standard television video picture in this case, it is possible to combine the displays. The application of this $225 module (lower prices in larger quantities) provides ample opportunities for the imagination.■

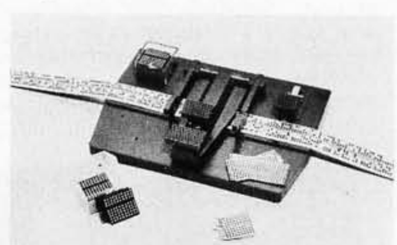
Circle 621 on inquiry card.

Ruggedized Power Supplies

Calex Manufacturing Company Inc, 3305 Vincent Rd, Pleasant Hill CA 94523, has come out with this sealed power supply black box which is designed for use with small dedicated microprocessor systems using the 8080. The supply transforms an AC input at 110 V (or several other voltages) into +12 V at 225 mA, –5 V at 20 mA, and +5 V at 1.25 A. These voltages are sufficient to run a typical 8080 system with 8080, 8024, 8228, two 2708s and several programmable memory parts with about 500 mA of TTL logic power left over on the +5 V lines.■

Circle 622 on inquiry card.

A Splice in Time Saves Nine?



Master Digital Corporation, 1308-F Logan Av, Costa Mesa CA 92626, has come out with this $60 tape splicing jig which can be had with various options for tapes from 5 to 8 channel widths. The company also makes pressure sensitive mylar tape patches in lengths from one to 12 inches, opaque or clear.■

Circle 623 on inquiry card.

Listing 1.

Location	Code	Key	Commentary
000	01	1	
001	51	SBR	Deal card to dealer
002	32	SIN	
003	42	STO	
004	00	0	Store as face down card.
005	03	3	
006	02	2	
007	51	SBR	Deal card to player.
008	32	SIN	
009	01	1	
010	51	SBR	Deal card to dealer.
011	32	SIN	
012	41	GTO	Jump around subroutine
013	12	CB	
014	46	*LBL	Begin "Deal Card" subroutine
015	32	SIN	
016	42	STO	
017	00	0	
018	00	0	
019	22	INV	
020	57	*fix	
021	43	RCL	
022	09	9	
023	09	9	
024	40	*x^2	
025	52	EE	Generate random number in R99
026	00	0	
027	09	9	
028	22	INV	
029	52	EE	
030	42	STO	
031	09	9	
032	09	9	
033	75	–	
034	53	(	
035	42	STO	
036	55	o/o	
037	01	1	
038	03	3	
039	75	–	
040	93	.	
041	05	5	
042	54	)	
043	57	*fix	Convert random number to a card value.
044	00	0	
045	52	EE	
046	22	INV	
047	52	EE	
048	65	X	
049	01	1	
050	03	3	
051	85	+	
052	02	2	
053	75	–	
054	42	STO	
055	00	0	Store card value
056	05	5	
057	01	1	
058	01	1	
059	95	=	Is card = 11 [ACE] ? If so then go to COS
060	90	*if z	
061	33	COS	
062	22	INV	
063	80	*if p	Is card in range 2 to 10? If so then go to TAN.
064	34	TAN	
065	01	1	
066	00	0	Force card value of 10 if outside legimate range
067	41	GTO	
068	65	X	
069	46	*LBL	
070	33	COS	
071	36	*IND	
072	43	RCL	
073	00	0	
074	00	0	Blackjack Logic:
075	75	–	Would hand go over 2, if ace counted as 11?
076	01	1	If so then ACE = 1 else ACE = 11
077	01	1	
078	95	=	
079	22	INV	
080	80	*if p	
081	34	TAN	
082	01	1	

SR-52 Card BLACKJACK

Michael J Garvey, vice president and systems consultant with General Computer Services Corporation, 2308 Central Av, Middletown OH 45042, sends in a game program for the SR-52, which was accompanied by this note:

I appreciate your article in your December issue on the "buried gold" in

Operating Procedure For "SR-52 Card BLACKJACK":

1. Load the program card, both sides, after it has been prepared.
2. Prime the random number generator with a 9 or 10 digit number as its seed. A good choice of a priming number is the current time of day (24 hour clock), followed by the date. This gives 10 digits total in the format.
   Enter: hhmmyymmdd
   Then press: STO, 9, 9
3. Enter the amount of your bet and start the game:
   Enter: bet amount
   Then press: A
4. Outcome:
   If the display flashes, then the game is over. The display shows the total of dealer and player hands.
   Press: CE to stop flashing.
   Then press: RCL, 9, 8 to read out the cumulative score if desired.
   Go to Step 3 to restart game.
   If the display is not flashing, the dealer's face down card is not shown.
5. Player options:
   To take a "hit,"
   Press B
   Then go to step 4.
   To "stand,"
   Press C
   Then go to step 4.
   To "double down" (if player's hand is 10 or 11 and not initial deal).
   Press D
   Then go to step 4.

Notes on operation of the game:

The player always wins at 21.

House always draws to 16 or lower, stands on 17 or higher.

On a "push" (both hands equal) neither side wins unless player has 21.

Listing 1, continued:

Location	Code	Key	Commentary
083	46	*LBL	Store adjusted card value.
084	65	X	
085	42	STO	
086	00	0	
087	05	5	
088	46	*LBL	Add card to total hand of receiver.
089	34	TAN	
090	43	RCL	
091	00	0	
092	05	5	
093	36	*IND	
094	44	SUM	
095	00	0	
096	00	0	
097	56	*rtn	
098	46	*LBL	Game entry (start here)
099	11	A	
100	47	*CMs	Clear bet in R19
101	42	STO	
102	01	1	
103	09	9	
104	86	*rset	Go to 000
105	46	*LBL	"Double down" entry
106	14	D	
107	02	2	Multiply bet by 2
108	49	*PROD	
109	01	1	
110	09	9	
111	50	*st fl	Turn on "stand" flag
112	00	0	
113	46	*LBL	"Hit" entry
114	12	B	
115	02	2	Deal card to player
116	51	SBR	
117	32	SIN	
118	41	GTO	Skip around "stand"
119	45	y^x	
120	46	*LBL	From step 012
121	13	C	
122	50	*st fl	Turn on "stand" flag
123	00	0	
124	46	*LBL	If player's total = 21 then go to 2' else if player's total >21 then go to 1'
125	45	y^x	
126	43	RCL	
127	00	0	
128	02	2	
129	75	–	
130	02	2	
131	01	1	
132	95	=	
133	90	*if z	
134	88	*2'	
135	80	*if p	
136	87	*1'	
137	43	RCL	If dealer's total = 21 then go to 1' else if dealer's total >21 then go to 2'
138	00	0	
139	01	1	
140	75	–	
141	02	2	
142	01	1	
143	95	=	
144	90	*if z	
145	87	*1'	
146	80	*if p	
147	88	*2'	
148	22	INV	If "stand" switch on then go to 8' [to display]
149	60	*if fl	
150	00	0	
151	68	*8'	
152	43	RCL	Is dealer's total $\geq$17?
153	00	0	
154	01	1	
155	75	–	
156	01	1	
157	07	7	
158	95	=	
159	80	*if p	If so then go to 3'
160	89	*3'	
161	01	1	Draw another card for dealer
162	51	SBR	
163	32	SIN	

the Texas Instruments SR-52 Programmable Calculator; it confirmed my suspicions that my SR-52 had more power than the instruction manual said.

Enclosed is a program listing for a program that I have written that will allow you to play BLACKJACK with an SR-52. This program was the toughest that I have written for that machine, since the 224 program steps allowed just didn't seem enough for the game; several days were spent in working and reworking the code in order to get the game to fit with the features I wanted. As you can see, it just fits, exactly.

I have sent this program to you in case any of your readers would be interested in it. My family and friends have had a great deal of fun with it, and it's especially great for killing time on a long trip; one person can "stake" the "house," while another person is the player. The program automatically keeps score for the player, and even handles the "double-down" feature of the game.

I submit it for the entertainment of anyone who wants to use it.

Listing 1 shows the program code, which we typeset using column headings from the original form, along with the register allocations. The procedures for using the SR-52 Card BLACKJACK program are summarized in the box labelled "Operating Procedure."

Location	Code	Key	Commentary
164	41	GTO	Go to C
165	13	C	
166	46	*LBL	
167	89	*3'	
168	43	RCL	Is dealer's total = player's total?
169	00	0	
170	01	1	
171	75	–	
172	43	RCL	
173	00	0	
174	02	2	
175	95	=	
176	90	*if z	If so then go to 7'
177	67	*7'	
178	22	INV	Is dealer's hand less than player's
179	80	*if p	
180	88	*2'	If so go to 2'
181	46	*LBL	Otherwise, dealer wins
182	87	*1'	
183	01	1	So make bet amount negative.
184	94	+/–	
185	49	*PROD	
186	01	1	
187	09	9	
188	46	*LBL	Add current bet to player's total score
189	88	*2'	
190	43	RCL	
191	01	1	
192	09	9	
193	44	SUM	
194	09	9	
195	08	8	

Listing 1, continued:

Location	Code	Key	Commentary
196	46	*LBL	
197	67	*7'	
198	43	RCL	
199	00	0	Game over, so display end game
200	03	3	
201	85	+	
202	85	+	
203	46	*LBL	
204	68	*8'	
205	57	*fix	
206	02	2	
207	43	RCL	
208	00	0	
209	01	1	
210	75	–	
211	43	RCL	
212	00	0	Display dealer hand as integer, player as decimal, and await next round.
213	03	3	
214	85	+	
215	43	RCL	
216	00	0	
217	02	2	
218	55	o/o	
219	01	1	
220	00	0	
221	00	0	
222	95	=	
223	81	HLT	

The SR-52.

ALLOCATIONS FOR SR-52 CARD BLACKJACK:

User accessible labels:

A = Start game
B = Hit
C = Stand
D = Double

Flag Usage:

0 = "stand" flag

Register Utilization:

00 = pointer for subroutine parameter
01 = dealer count
02 = player count
03 = facedown
05 = current card
98 = winnings
99 = random number output
19 = current bet ■

A Guide to Baudot Machines:

Part 3, A Teleprinter Test Circuit

Michael S McNatt
4658 E 57 St
Tulsa OK 74135

Now that you've found out what type of Baudot teleprinters are available on the surplus market, and where to go to get information on how to interface them to your microprocessor, it might be convenient to build a test box to check the working condition of your new acquisition. What follows is a circuit for just such a test box which can be used to provide the 60 mA current loop required by the Baudot machine. Circuits are also included in the box to generate signals which can verify correct machine function. Although not as handy as the test box, a Baudot keyboard may be used to test a page printer of the same speed. The test box has variable control of data rate for testing all Baudot teleprinters. The test box circuits supply the following functions:

1. **Standard RY test signals**, either continuously or at two second intervals, for mechanical alignment purposes. These signals result in a maximum amount of slipping and sliding of adjacent gears and parts within the machine, conditions which are also most likely to cause malfunctions to surface. The two second interval used with the RY test prevents a rather large waste of paper or tape during a long test.
2. **Individual Baudot characters** at two second intervals. Five bit-switches select the particular character desired (see table 1 in part 1 of this article for bit codes). One use of this function is to check various machines to see which keys are actually installed for selected bit combinations. For example, there are at least three different "figures" code assignments. Another use is in troubleshooting, for instance when the wrong character is appearing on the printer or punch in response to a processor output command. The test box can be quickly switched from "CPU" to local "char" mode to verify that the source of the problem is or is not the machine itself.

Also included in the test box is circuitry to accept serial TTL or CMOS level Baudot coded output from the processor or hardware code converter. The "MODE" switch

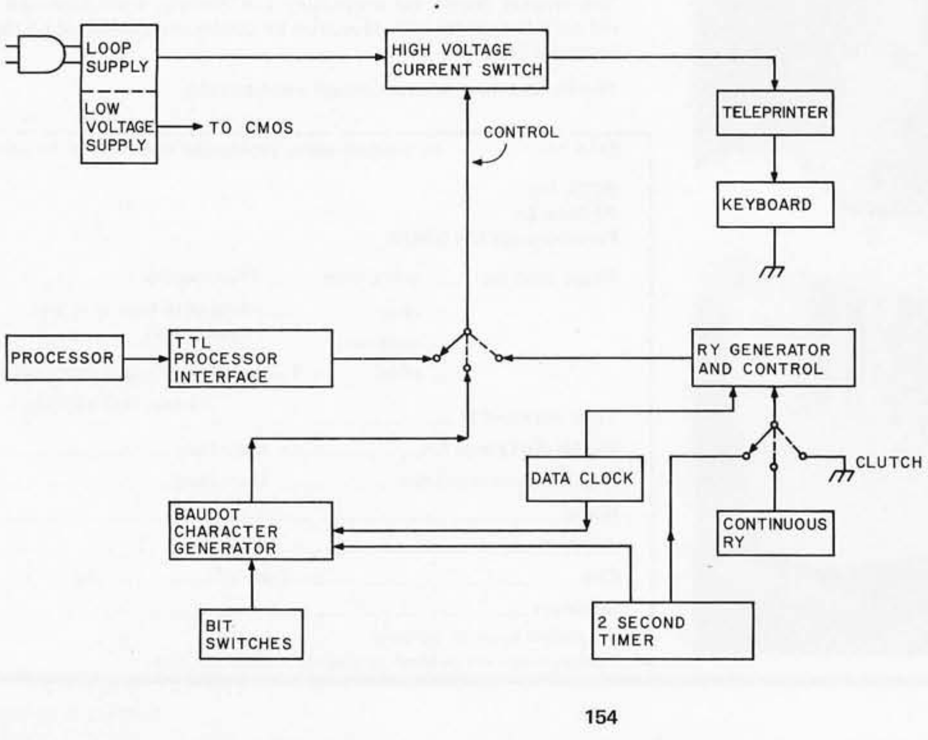
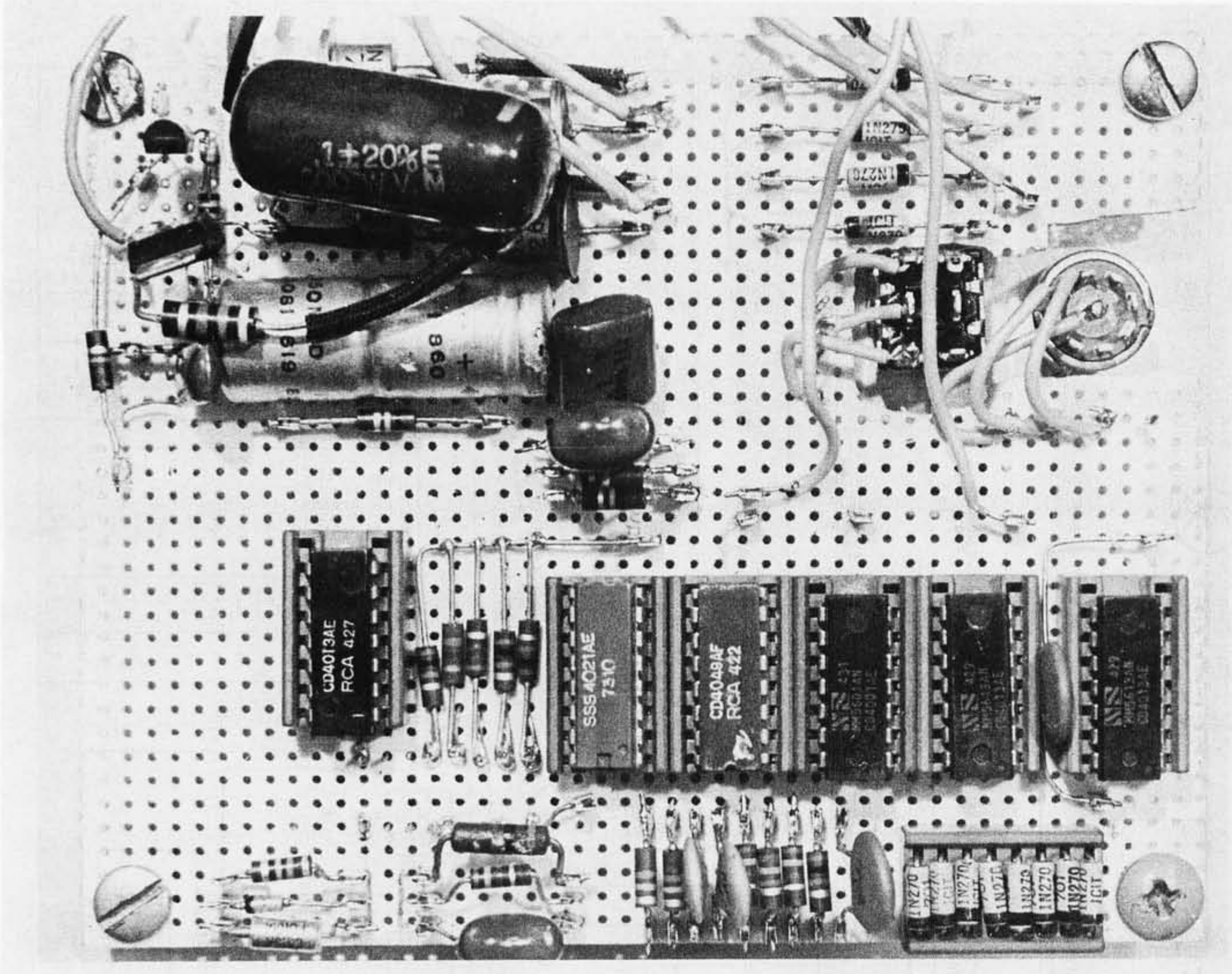


Figure 1: Block diagram of a Baudot machine test box allowing the generation of continuous or intermittent 'RY' test signals, or individual Baudot characters.



selects between processor input, RY test signals, and the individual character generator.

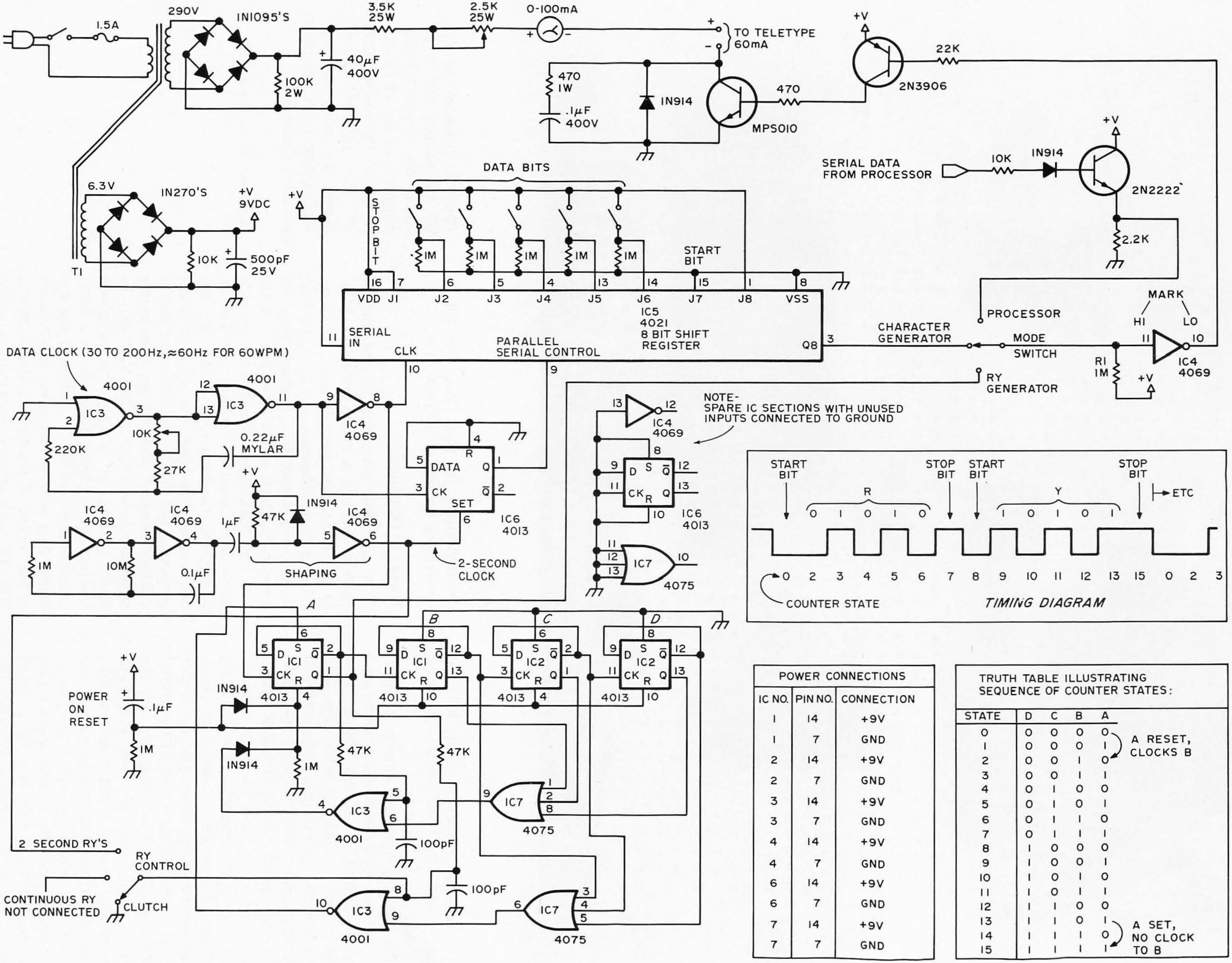
The absolute minimum amount of hardware one can get by with when setting up a Baudot machine is a high voltage loop supply. Some machines, Model 28s and Kleinschmidts, for example, come equipped with this supply; unfortunately, my Model 15 RO page printer didn't. It may be a good idea to borrow or build a test box anyway. When trying to determine the operating condition of a prospective Baudot machine, taking along a simple loop supply can enable at least a very minimal check of the printer mechanism. When purchasing my 15, the seller, a ham, used his own supply to demonstrate successful clutch action, then opened and closed the loop by rubbing the connections, causing random characters to be printed. Obviously, a better check would have been with a test box or a speed compatible Baudot keyboard. However, the demonstration, in this case, was totally satisfactory because of the price, $30!

Theory of Operation

A complete schematic of the Baudot machine test box is shown in figure 2. Many of the parts used were selected solely because they were immediately available from the proverbial junk box. (I think there are companies which would envy some of the so-called "junk boxes" I've seen belonging to hams and computer phreaks . . .) For instance, the transformer shown was the only one usable that came out of a thorough purging of not only my junk box but that of a fellow designer. Actually any transformer secondary from 110 V to 300 VAC may be used, as long as adequate series power resistors are included to result in a short circuit current of 65 to 70 mA. Telegraph loops, such as Western Union, usually work at 260 V. Use of the higher voltages, around 200 VDC, although requiring more power to

Photo 1: A sample layout of the Baudot test box. Note the eight diodes plugged into the 16 pin IC socket at the lower right of the picture. These diodes are being used in replacement of IC7 in the circuit of figure 2.

Photo courtesy of Don Clum



POWER CONNECTIONS		
IC NO.	PIN NO.	CONNECTION
1	14	+9V
1	7	GND
2	14	+9V
2	7	GND
3	14	+9V
3	7	GND
4	14	+9V
4	7	GND
6	14	+9V
6	7	GND
7	14	+9V
7	7	GND

TRUTH TABLE ILLUSTRATING SEQUENCE OF COUNTER STATES:

STATE	D	C	B	A	
0	0	0	0	0	A RESET, CLOCKS B
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	
10	1	0	1	0	
11	1	0	1	1	
12	1	1	0	0	
13	1	1	0	1	A SET, NO CLOCK TO B
14	1	1	1	0	
15	1	1	1	1	

←

Figure 2: A Baudot machine test box and interface. This circuit generates 5 level Baudot code to be used in testing the working condition of a Baudot teleprinter. All resistances are measured in ohms, and unless otherwise marked are 1/4W and ±10% accuracy. The circuit allows either continuous RY test strings or intermittent 2 second test strings. The RY test string causes the most movement of gears and will show any difficulties in operation quicker than other test string combinations. All unused inputs of any CMOS integrated circuit must be grounded.

be wasted in the series resistors, results in a faster response in the printer selector magnets, which is especially important at the higher speeds such as 100 words per minute. Do *not* put the standard transient suppression diode across the coils because selector magnet release will be slowed down.

Since CMOS logic elements are used, the low voltage secondary winding can be of any voltage which will result in from 5 to 12 VDC for the +V supply. The series resistor capacitor network across high voltage transistor Q1 is used to suppress the voltage transient occurring in the magnet coils when Q1 is turned off. A low logic state at IC4, pin 10, results in a mark or clutch state, since this will switch Q2 and thus Q1 on, causing loop current to flow. Resistor R1 tied to +V insures that the machine will remain in the mark state while changing the MODE switch.

The data clock and the 2 second interval clock, IC3 and IC4, are standard CMOS oscillators. In substituting CMOS, try to avoid 4049 inverters in these oscillators. The character generator is formed by 8 bit shift register IC5, which is parallel loaded with the 5 Baudot data bits, a start bit, pin 15, and a stop bit, pin 7. The remaining bit, pin 1, is tied high. Characters are generated every 2 seconds when the load control, pin 9, is pulled high for one half period of the data clock. Flip flop IC6 synchronizes the load pulses from the slow interval clock with the much faster data clock. Note that serial input IC5, pin 11, is tied high. This provides marking 1s to be introduced into the shift register after the parallel loaded Baudot bits have been shifted out.

The RY generator is essentially a CMOS version of an RTL circuit published in *Ham Radio* magazine, March 1971, pages 23 to 29. Briefly, IC1 and IC2 form a four bit, labeled A through D on schematic, binary counter which skips two states when counting from 0 to 15. The RY pattern thus generated is shown in the timing diagram of figure 2. Referring to the schematic, counter state 1, DCBA = 0001, causes flip flop A to be reset, thus clocking flip flop B. This occurs when IC1, pin 2, goes high, causing the reset pulse at IC3, pin 4. Counter state 14, DCBA = 1110, causes flip flop A to be set, with no clocking of flip flop B. This occurs when IC1, pin 1, goes high, causing the set pulse at IC3, pin 10.

The RY control is a three position SPDT minitoggle switch with the center position off. A neat feature of this circuit is that no matter when the RY control is switched from one position to another, complete RYs are always generated.

Miscellaneous Notes

- I used six 1N914 diodes and two resistors in place of IC7, hence the 16 pin socket was used as a diode holder, (see photo 1). The 4075s are more convenient, but the diodes were in the junk box at the time. If diodes are used, connect three of the anodes in place of IC7, pins 1, 2, 8, and the cathodes in place of IC7, pin 9, and tie a 1M resistor to ground at this point. Ditto to replace the second gate.
- To avoid the self-smoke or reach for the sky mode, all unused CMOS inputs *must* be grounded or tied to +V.
- Obviously the test box can be as simple or complex as one desires. The minimum configuration would probably be two Eveready No. 416 67.5 V batteries and a resistor in series, using the "sparking" method to generate random characters!
- Seriously, the RY generator is mandatory, with the character generator and 2 second timer as options. Of course a more sophisticated Baudot test box would have an 8080A, a 1702A, an 8251, and maybe some hexadecimal character readouts . . .

Summary

An old surplus Baudot code teleprinter is the most inexpensive hardcopy peripheral available to the computer hobbyist today. This series of articles has presented approaches to acquiring, using and testing these units, as well as sources of reference material and interfacing hardware and software. It is hoped that the information provided will greatly ease the acquisition and interfacing tasks facing the hobbyist who owns or is planning to own one of these practically indestructible machines.■

What's New?

GNAT 3M Drive

This new GNAT MC-200 data storage system will make an interesting option for many users. This is an RS-232 interfaced serial device which plugs into a standard DB-25 connector often used

with RS-232. It gives the user a serial storage capability on 3M DCD-100 cartridges. Optional parallel IO is also available.

The unit has sufficient intelligence built in to recognize various command sequences including start write, start read, stop write, stop read, rewind, etc. All these sequences are duplicated by hardware switches on the front panel for manual control as well. The data rate for the device, which looks logically like any RS-232 serial terminal, can be set by a switch at rates from 75 to 9600 bps. Options available include parallel interface, file search capability, dual drives, etc. The price of this system is $1930, and its mix of features should prove quite useful for those who want a file storage system with the minimum of interface complexity.

GNAT is located at 7895 Convoy Ct, Unit 6, San Diego CA 92111, and delivery is quoted as from stock to 60 days ARO.■

Circle 613 on inquiry card.

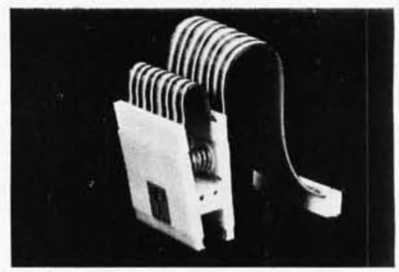
The Ultimate in Terminal Printers

The ultimate in terminal printing mechanisms is that provided by the Diablo HyType II printers and similar high-speed, high quality impact mechanisms. These mechanisms give print quality (with carbon ribbons) good enough to photocopy and reproduce, yet at character printing rates of up to 45 characters per second. Applied Computer Systems, 248 Sobrante Way, Sunnyvale CA 94086, has taken the Diablo mechanism and placed it into this attractive package along with a microprocessor and numerous options. The result is a hard copy oriented microcomputer with prices starting at $4500 and options including floppy disks, memory to 64 K, plotting, sort and merge capability, down loading of programs over communications lines to large systems, and user programmability of the built-in microprocessor. Communication speeds of 600, 1200, 2400 and 4800 bps are supported, and the standard memory size is 4 K bytes. Keyboards customized for APL, ASCII and IBM 2741 compatibility are available.■

Circle 612 on inquiry card.

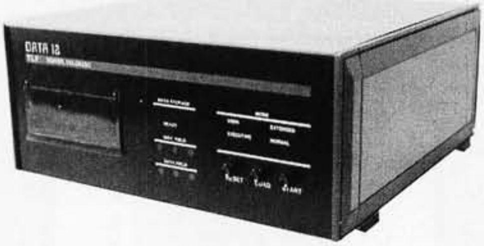
An Interesting Test Fixture

AP Products, POB 110, 72 Corwin Dr, Painesville OH 44077, has introduced this interesting new fixture for use in testing integrated circuit assemblies. Called the "Logical Connection," this is a preassembled ribbon cable attached to an IC test clip at one end, and a flat cable socket connector at the other. The length of the ribbon cable can be chosen by the purchaser; the example here is of course rather short to illustrate the idea.■



Circle 614 on inquiry card.

An Interesting New Product for People Wanting Complete Systems

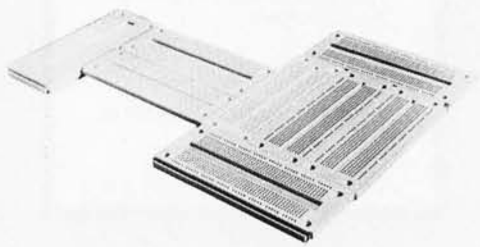


Frank Laczko, president of TLF, POB 2298, Littleton CO 80161, called us early in February to announce his new brainchild, the Data 12 computer, shown here. This machine is an example of a completely integrated system oriented to a terminal user. Its architecture is that of the Digital Equipment Corporation's PDP-8E, using the Intersil IM6100 microprocessor.

The $1695 price tag of this processor with its built-in tape drive gets a PDP-8 like computer with 4096 12 bit words of user memory, serial terminal interface, tape controller with one drive built in, and a tape operating system that includes both an unattended batch mode of operation and real time task scheduling capability. The random access tape cassette drive uses a preformatted digital cassette that has an average access time quoted by the TLF press release as less than 25 seconds, with bidirectional search speeds of 100 inches per second. The tape holds a maximum of 262,144 words in 128 word blocks. The software supplied with the system is completely oriented to an interactive keyboard, and it includes an "invisible" system executive that handles all IO scheduling, buffering and vectoring. The operating system, modeled after DEC's OS-8, is written with an eye towards device independence in the same fashion as larger computers. The system software manages named files for mass storage control, and also includes the usual text editor for program preparation, a symbolic assembler, dissassembler and loaders. The system is marketed with a BASIC compiler which includes multisegment program linkage conventions, large multidimensional arrays, string handling, and multiline user defined functions. User memory can be expanded to 32 K words, and additional peripherals are also available.■

Circle 611 on inquiry card.

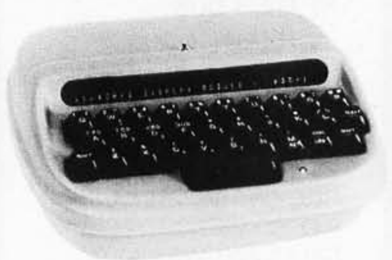
A New Variation on Solderless Prototyping Boards



Continental Specialties, 44 Kendall St, POB 1942, New Haven CT 06509, has announced an interesting new variation on the solderless breadboard concept: modular units with a set of interlocking edges so that they can be built up into multiboard arrays that are rigidly held together.■

Circle 609 on inquiry card.

A Portable Display Terminal



The Micon KDM/1 is manufactured by Micon Industries, 252 Oak St, Oakland CA 94607. The terminal is a self-contained keyboard with 32 character alphanumeric LED display, available for $400 mail order. The product is available in eight different colors. It is also available in many fine computer stores. Options include acoustic coupler, digital cassette tape storage and additional memory capacity to 1920 characters.■

Circle 610 on inquiry card.

BYTE's Bugs

Oops . . . Some Phi-Deck Updates to Freeman's Article

I have read an article published in the March 1977 issue of BYTE. The article, "Cassette Transports for the 'Roll Your Own' Hobbyist," had some errors in it concerning the Phi-Deck cassette transports. These errors are apparently due to the normal time lags involved between the occurrence of a development and the time consumed to communicate the event.

Areas I would like to bring to your attention are:

Solenoid Operation

Phi-Deck uses a 4 motor design for all control functions. No solenoids are used, thus eliminating the power con-

Floating Point Software for the 8080

PACKAGE INCLUDES MANUAL, OBJECT CODE ON PAPER TAPE, AND COMPLETE ANNOTATED SOURCE LISTING.

ROUTINES PERFORM:
- Add, subtract, multiply, and divide
- Load and store floating point accumulator
- Format conversion
- Square root
- Sine and cosine
- Natural logarithm and exponential
- Arc tangent
- Hyperbolic sine and cosine
- Logarithm base ten

OTHER FEATURES:
- Compatible with any 8080 microcomputer
- Requires less than 2K bytes program store
- High speed--worst case multiply 2.5 msec.
- Accurate to six decimal digits
- Low cost (source tape available separately)

SEND $10 TO:

BURT HASHIZUME
P.O. BOX 447
MAYNARD, MASS. 01754

Circle 262 on inquiry card.

MICRO COMPUTERS

ORANGE COUNTY

Hardware and Software
SERVICE and SALES

Fountain Valley Plaza
½ Block South
of
San Diego Freeway

18120 Brookhurst St.
Fountain Valley
Calif. 92708
714-963-5551
• Hours •
Monday-Friday 1-9
Sat 10-6 Sun 12-5

Circle 261 on inquiry card.

THE BETTER BUG TRAP

The Better Bug Trap is an Altair/IMSAI plug-compatible board that extends system capabilities to facilitate software debugging and real-time processing. Capabilities include interval timer, real-time clock, watchdog timer, processor slowdown, and clock with variable rates. Four hardware breakpoint addresses allow you to stop processing or generate an interrupt at a breakpoint without modifying existing software. The board services its interrupt with a CALL instruction to ANY memory address you choose. All capabilities may be set by software or front panel. Write for free literature.

$180 assembled, tested, complete documentation, software.

MICRONICS, INC.

PO Box 3514, Greenville, N.C. 27834
919 - 758 - 7757

Circle 196 on inquiry card.

COMPUTER CLUBS
PROFESSIONALS-AMATEURS
BUSINESSMEN-HOBBYISTS

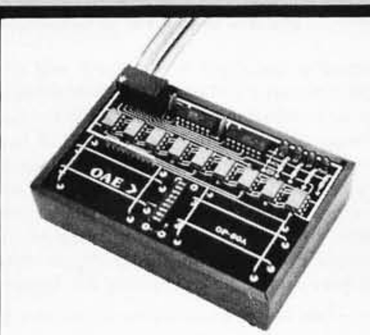
ORGANIZE (5-20 PEOPLE) AND BUY A FULL TELEPROCESSING SYSTEM (Z-80 MICROCOMPUTER, CUSTOM BOARDS, DUAL FLOPPY). THEN BUY SINGLY REMOTE OR IN-HOUSE VIDEO TERMINALS (WE SELL THESE TOO!). ADD MVT's BOSSYSTEM/1 SOFTWARE AND YOU HAVE A "DEDICATED" COMPUTER FOR 5-20 USERS.

USING OUR MULTI-USER BASIC INTERPRETER YOU ARE NOW READY FOR EASY REMOTE PROGRAMMING, PROCESSING, GAMES, ETC. FOR A COMPLETE SYSTEM (TESTED AND BURNED-IN WITH SOFTWARE, TERMINALS AND PERIPHERALS) THE PRICE FOR 10 PEOPLE IS LESS THAN $1900 PER PERSON.

WRITE NOW FOR FURTHER DETAILS ON THIS NEW AND EXCITING DIMENSION IN PERSONAL COMPUTING.

MVT MICROCOMPUTER SYSTEMS, INC.
P.O. BOX 62
AGOURA, CALIFORNIA 91301

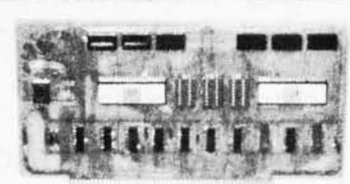
Circle 264 on inquiry card.



Pictured above is the new OP-80A High Speed Paper Tape Reader from OAE. This unit has no moving parts, will read punched tape as fast as you can pull it through (0-5,000 c.p.s.), and costs **only $74.50 KIT, $95.00 ASSEMBLED & TESTED.** It includes a precision optical sensor array, high speed data buffers, and all required handshake logic to interface with any uP parallel I/O port.

To order, send check or money order (include $2.50 shipping/handling) to Oliver Audio Engineering, 7330 Laurel Canyon Blvd., No. Hollywood, CA 91605, or call our 24 hr. M/C-B/A order line: (213) 874-6463.

Circle 64 on inquiry card.



SIX PORT I/O CONTROLLER
- Program Control of Ports
- S100 Buss Compatible
- Parallel I/O
- Single Printed Circuit Board

VIDEO SYSTEM
- 30 x 64 Display – 1920 Char.
- 120 Character Buffer
- 1 μsec Read/Write Time
- Extensive Software Provided
- Display Attribute Controls Are All Programmable

Available in Kits or Assembled
BankAmericard and Master Charge Accepted

Dealer or Club Discounts Available

IOR
Box 28823 Dallas, Texas 75228

Circle 268 on inquiry card.

sumption associated with maintaining a solenoid in the active state.

Die Cast Chassis

All Phi-Deck models now utilize a precise die cast chassis. Triple I no longer uses the sheet metal frames.

AC Phi-Deck Models

Triple I is announcing two new Phi-Deck transports which will give us a total of five models. Both of the new models will use an AC capstan motor.

New Pricing

Phi-Deck Models 1 and 2 are now built with the die cast chassis. This and other factors have increased the price of these models to $124 in quantities of 1 to 9. The price is higher, but you can now get a precise die cast chassis for under the $169 mentioned.

If you could update your audience about the above corrections to the article, it would be greatly appreciated.

Jack Morrow
Applications Engineering Technician
Triple I
POB 25308
Oklahoma City OK 73125

My fault. In editing, I should have correlated later Phi-Deck literature with the article, which was written in the Summer of 1976 . . . CH ■

Advanced Microcomputer Products Presents

CONTINENTAL SPECIALTIES CORPORATION

CSC Prototype Design Equipment

QT SOCKETS AND BUS STRIPS

Now in stock for Microprocessor Prototyping
NEW SOCKETS & BUS STRIPS

Experimentor 600 $10.95
.6 spacing for microprocessor applications

Experimentor 300 $9.95
.3 spacing for microprocessor applications

PROTO-BOARD 100 KIT · PROTO-BOARD 101 · PROTO-BOARD 102 · PROTO-BOARD 103 · PROTO-BOARD 104 · PROTO-BOARD 203 · PROTO-BOARD 203A $120.00

LOGIC PROBE LP-1

WE ARE A FULL LINE CSC DISTRIBUTOR. WE CAN SUPPLY ANY PRODUCT OUT OF THE CSC CATALOGUE. IF YOU NEED THEIR LATEST CATALOGUE WRITE AND WE'LL SEND IT ALONG.

8K RAM BOARD $198.00

ADVANCED ★ MICROCOMPUTER PRODUCTS

BUILD YOUR OWN COMPLETE COMPUTER SYSTEM

★ AMI EVK 99 6800 BASED MICROCOMPUTER

Ask the Home Brew Computer Club in Santa Clara, California why they choose the EVK 99 Microcomputer. They now have 85 systems. The reasons are clear. Performance, versatility and cost effectiveness. Photo is of expanded EVK 99 Board

Photo of expanded EVK 99
NOW $133.00

1. **EVK 99** consists of the following only
   - Qty (1) PCB 10½ x 12
   - (1) 6800 8 bit CPU
   - (1) 6850 ACIA (Asynchronous Communication Interface Adapter)
   - (1) 6820 PIA (Peripheral Interface Adapter)
   - (2) 6830 ROM Subroutine Program Library (1) 6831 16K ROM
   - (4) 6810 1 RAM
   - (1) Operating Manual & Instructions

   To implement fully expanded system additional IC's are required. (See Photo)
2. **UNIVERSAL KLUGE BOARD $98.00**
   Completely compatible to the EVK system. This board allows the user to add any additional parts he desires. It has 5400 holes, holds 94 16 pin IC's or others on .2, .4, .6, .9" or greater centers. 2 - 50 pin, 1 - 20 pin flat cable, 1 - 25 pin RS232 connector
3. **16K BYTE RAM BOARD $77.95**
   16K x 8 or 8K x 16 organization. Uses 2102's or 21L02's and adapts to slow memory
4. **6 SLOT MOTHER BOARD $29.95**
   Connectors mount on .75" centers. May be plugged together for expanded applications
5. **EXTENDER BOARD $46.50**
   For use with back plane above in card cage configuration.
   New boards are now in development including cassette I/O and others. If you want more information drop us a line.

★ MICROPROCESSOR PRODUCTS

8080A/9080A	$24.95	S6860			$15.95			21L02/91L02	1.99
AMI S6800	24.95	S6820			11.00			2102/9102	1.89
AM2901	29.95	S6810-1			6.25			9111/2111	4.75
GI CP1600	69.95	S6850			10.95			9101/2101	5.25
Z80	69.95	S2350			12.95			9112/2112	2.95
6502	29.95	S1883			5.90			2489	1.88
AM2902PC	$3.95	S6834-1			22.95			4104/4804	12.95
AM2905PC	10.95	S6800			15.00			4402A	12.95
AM2906PC	11.95	S6800			15.00			4200	12.95
AM2907PC	10.75				22.95			9131APC	14.95
AM2909PC	24.95				175.00			9141APC	14.95
AM2911PC	18.95	8212	$4.75	AM2533	7.50			9050/TMS4050	12.95
AM2918	6.75	8224	6.25	AM2401	16.50			9060/TMS4060	12.95
AM29720PC	7.95	8216	4.95	AM1402	6.95			P3101	3.25
AM29721PC	7.95	8226	5.95	AM1403	5.95			5101	11.95
AY 51013A	5.50	8228	9.90	AM1404	6.95			1702A	10.90
AM2812	14.95	9551	19.95	MM5006H	2.25			2708	68.00*
AM2841	12.00	9555	19.95	MM5016H	4.95			5204	23.50
AM3341	9.95			AM3347	6.40				

TV GAME KITS

Qty (1) GI AY38500-1 TV Game Chip ONLY $29.95

TV KIT NO.1 ONLY $34.95
Includes AY38500-1 TV GameChip. PC Board for video composite output plus instructions. Add $5.95 for 2MHz crystal

TV KIT NO. 2 ONLY $49.95
Includes AY38500-1 TV GameChip. PC Board, Switches, pots, speaker, crystal, XFMR, Bridge and all parts required for video composite. Works super with terminal - includes instructions and data sheets

SPECIAL
BUY 10 OF THE FOLLOWING AND YOUR PRICE IS. . .

AMD 8080A	$19.95
AMD 9111	4.10
AMD 9112	2.30

RF MODULATOR MODULE

To convert your video composite to RF try UM 1021 Channel 3/4 VHF Modulator $8.95

- DC Switchable VSWR > 2 max VCC = 6.5 volts
- DC Coupled modulation input with positive transfer characteristic. Designed for use with TV Games

SPECIALS

75452	10/2.50
8T26B	2/5.00
74LS138	1.49
8097	.99
74166	.80
74161	.49
741HC	10/2.50
MH0026CN	3.95
1488	1.29
1489	1.29

★ NEW PRODUCTS

F4702 Baud Rate Gen.	15.95
8214 Priority Int.	11.95
Z80 CPU	49.95
Z80 3881 PIC	9.95
Z80 3882 CPC	9.95
PD 372 Floppy Controller	55.00
PD 371 Mag. Tape Controller	49.95
APP Notes for 372 Floppy	6.95
GI AY5 3550 4½ Digit DVM	29.95
GI AY5 3600 Keyboard Encoder	14.95
Intersil 8038CC Wave Gen.	3.50
AY3507 DVM Chip	12.95
RO3 2513 Uppercase	9.95
RO3 2513 Lowercase	10.95
TMS 4060 NL-2 4K RAM	3.95
MK4096 4K RAM	3.95
1602B UART	5.50
1883 UART	6.50
AY51013A UART	5.50
5101C-E 1K CMOS RAM	6.95
WD 1671B Astros	29.95
WD 1941 Dual Baud Gen	9.95
WD 1771 Floppy Disc Cont.	74.95
6330-1 256 Bit OC Prom	2.95
6331-1 256 Bit TS Prom	2.95
6300-1 1K OC Prom	3.50
6301-1 1K TS Prom	3.50
6305-1 2K OC Prom	9.95
6306-1 2K TS Prom	9.95
6340-1 4K OC Prom	18.95
6341-1 4K TS Prom	18.95
6352-1 4K OC Prom	18.95
6353-1 4K TS Prom	18.95

From Iasis David Guzeman's formerly of Intel - Complete 6 Volume Programmed learning course "Microcomputer Design is a Snap" only $99.50
Microcomputer Application Handbook only $7.95
Iasis 8080 Microprocessor System in a book complete instructional and Microcomputer
Send for complete details on all Iasis Products.

Intel MCS8 Users Manual	5.00
Intel MCS4 Users Manual	5.00
Raytheon Linear Catalogue	2.95
Raytheon Quads & Duals Manual	1.95

ALTAIR/IMSAI COMPATIBLE 8K MEMORY

Our new brain board comes complete with 21L02-1 memory chips (500ns) and sockets. All address control and data out lines fully buffered.
$198.00 COMPLETE
If you desire 91L02APC - Please add $9.95

★ IC MARKET PLACE ★

★ LINEAR

LM105	3.50	LM320K 5.2	1.45	LM381N	1.79
LM106H	2.50	LM320K 12	1.45	LM382N	1.79
LM300H	.80	LM320K 15	1.45	LM555CN	.58
LM301H	.35	LM320T 5	1.75	LM556CN	1.30
LM307CN	.45	LM320T 5.2	1.75	LM567CN	1.45
LM302H	1.25	LM320T 8	1.75	LM565N	1.95
LM304H	1.20	LM320T 12	1.75	LM565H	1.95
LM305H	.85	LM320T 18	1.75	LM703CN	.45
LM305AH	1.25	LM323K 5	[illegible]	LM709H	.40
LM306N	.95	LM324M	1.65	LM709N	.40
LM307CN	.35	LM339N	1.55	LM710N	.70
LM307H	.35	LM340K 5	1.95	LM711N	.39
LM308H	1.00	LM340K 8	1.95	LM711H	.39
LM308CN	1.00	LM340K 12	1.95	LM723N	.55
LM309H	1.10	LM340K 15	1.95	LM723H	.55
LM309K	1.25	LM340K 18	1.95	LM725H	3.25
LM310CN	1.15	LM340K 24	1.95	LM733N	1.00
LM310H	1.15	LM351CN	.65	LM733CH	1.45
LM311H	.90	LM370N	1.65	LM739N	1.29
LM311CN	.49	LM370H	1.65	LM741CH	.35
LM317H	1.95	LM373N	3.25	LM741CN 14	.39
LM318H	1.50	LM377N	4.00	LM741CD	.49
LM318CN	1.50	LM380N	1.39	LM747H	.75
LM319N	1.30	LM380CN	1.05	LM747N	.75
LM320K 5	1.45			LM748H	.39
				LM748N	.39

LM1414N	1.75	75324	3.50
LM1458CN	.65	SG4501T	2.40
LM1496N	.95	SG4501N	2.40
LM2111N	1.95	SG3524	4.95
LM2901N	1.95	RC4194T	6.95
LM3065N	.69	RC4195T	6.95
LM3900N	.55	RC1458CN	.89
LM3909	1.10	RC1556CN	.99
LM7520	2.25	RC3403AD	2.30
LM7524N	1.75	RC4131CN	.99
LM7525N	1.90	RC4136D	1.90
LM7534N	2.20	RC4558CN	.99
LM7535N	1.25	RC4151CN	6.50

★ CMOS

CD4000	.25	CD4026	1.95	CD4075	.35
CD4001	.25	CD4027	.85	CD4076	.35
CD4002	.25	CD4028	1.50	CD4081	.35
CD4006	1.75	CD4029	1.99	CD4082	.35
CD4007	.25	CD4030	.60	CD4510	1.95
CD4008	1.95	CD4033	2.49	CD4511	1.40
CD4009	.48	CD4034	4.95	CD4516	1.95
CD4010	.48	CD4035	1.79	CD4518	2.45
CD4011	.28	CD4040	2.10	CD4520	2.49
CD4012	.28	CD4042	1.45	CD4522	2.56
CD4013	.60	CD4043	1.45	CD4526	2.56
CD4014	1.95	CD4044	1.95	CD4527	2.70
CD4015	1.95	CD4049	.61	CD4528	1.50
CD4016	.59	CD4050	.61	CD4531	1.99
CD4017	1.25	CD4051	2.25	CD4543	2.25
CD4018	2.39	CD4066	1.20	CD4555	1.75
CD4019	1.55	CD4068	.35	CD4556	1.75
CD4020	1.35	CD4069	.35	CD4581	4.95
CD4021	1.85	CD4070	.99	CD4582	1.95
CD4023	.29	CD4071	.35	CD4585	2.45
CD4024	1.29	CD4072	.35	Also	
CD4025	.29	CD4073	.35	5101	$11.95

★ 7400 TTL

7400N	.18	7445N	.99	74122N	.44	74185N	$2.10
7401N	.18	7446N	.99	74123N	.60	74190N	$1.30
7402N	.18	7447N	.99	74125N	.79	74191N	$1.30
7403N	.18	7448N	.99	74126N	.79	74192N	$1.10
7404N	.19	7450N	.26	74132N	.70	74193N	$1.10
7405N	.20	7451N	.26	74141N	.88	74194N	.95
7406N	.25	7453N	.26	74145N	.70	74195N	.95
7407N	.25	7454N	.26	74147N	$1.59	74196N	.95
7408N	.21	7470N	.45	74148N	$1.29	74197N	.95
7409N	.21	7472N	.30	74150N	$1.15	74198N	$1.45
7410N	.21	7473N	.30	74151N	.98	74199N	$1.85
7411N	.21	7474N	.30	74153N	.98	74251N	$1.45
7412N	.21	7475N	.69	74154N	$1.15	74279N	$1.10
7413N	.45	7476N	.42	74155N	.95	8093N	.49
7414N	.69	7480N	.70	74156N	.95	8094N	.49
7416N	.35	7482N	.95	74157N	.95	8095N	.99
7417N	.35	7483N	.95	74160N	.99	8096N	.99
7420N	.19	7485N	.89	74161N	.99	8097N	.99
7421N	.35	7486N	.28	74162N	.99	8098N	.99
7423N	.35	7489N	$1.88	74163N	.99	75150N	$1.70
7425N	.35	7490N	.60	74164N	.99	75450N	$1.10
7426N	.25	7491N	.70	74165N	.99	75451N	.59
7427N	.35	7492N	.59	74166N	$1.10	75452N	.59
7428N	.28	7493N	.79	74170N	$4.20	75453N	.59
7430N	.21	7494N	.79	74172N	$16.25	75454N	.59
7432N	.31	7495N	.79	74173N	$1.40	75491N	.70
7433N	.30	7496N	.79	74174N	$1.25	75492N	.70
7437N	.35	7497N	$3.75	74175N	$1.25	75493N	$1.10
7438N	.35	74100N	$1.10	74176N	.79	75494N	$1.10
7440N	.21	74104N	.53	74177N	.79		
7441N	$1.05	74105N	.53	74180N	.79		
7442N	.69	74107N	.33	74181N	$2.75		
7443N	.69	74109N	.33	74182N	.79		
7444N	$1.50	74121N	.35	74184N	$2.10		

100 PIECES MIX TAKE 10% DISCOUNT
1000 PIECES MIX TAKE 15% DISCOUNT

PLESSEY SEMICONDUCTORS

RADIO COMMUNICATION

SL1610C	1.95	SL1624C	2.75
SL1611C	1.95	SL1630C	1.95
SL1612C	1.95	SL1640C	1.95
SL1613C	2.45	SL1641C	6.95
SL1620C	2.55	SL1645C	5.95
SL1621C	2.95	SL1650C	5.75
SL1622C	2.95	SL1651C	4.95
SL1623C	2.95	SL1680C	3.75

Application Manual for SL 600 Circuits 1.95
Plessey Semiconductor Catalogue 3.95

DIVIDERS

SP8600B	6.95	SP8620B	12.50	SP8640B	6.20
SP8601B	4.25	SP8621B	8.50	FSB641B	7.50
SP8602B	16.00	SP8622B	6.50	SP8642B	12.50
SP8603B	14.50	SP8630B	20.00	SP8643B	20.00
SP8604B	8.25	SP8631B	12.00	SP8646B	6.50
SP8607B	17.00	SP8632B	10.00	SP8647B	8.00
SP8613B	17.50	SP8634B	35.00	SP8650B	25.00
SP8614B	16.00	SP8635B	27.50	SP8651B	18.50
SP8615B	20.00	SP8638B	27.50	SP8652B	12.00
SP8616B	22.00	SP8637B	18.00		

6800 STARTER SET

$68.00

8080A STARTER SET

$80.00 — (1) 8080A 8 bit CPU; (1) 8212 I/O Port; (1) 8224 Clock; (1) 8228 System Controller; (24) 2102 RAM's

All shipments first class or UPS in U.S. or add 5% for shipping and handling. Minimum order $10.00. California Residents add 6% tax. Foreign add 8%.

ADVANCED ★ MICROCOMPUTER PRODUCTS
P. O. BOX 17329
IRVINE, CALIFORNIA 92713
PHONE (714) 968-3655

★ ORDER YOUR NEW CATALOG NOW !

Z-80 $49.95 4K STATIC RAM ★ 2708 $49.95

WIREWRAP SOCKETS - GOLD

14 Pin WW 1 level	.30
14 Pin WW 2 level	.33
14 Pin WW 3 level	.37
16 Pin WW 1 level	.31
16 Pin WW 2 level	.34
16 Pin WW 3 level	.38
18 Pin WW 3 level	.60
24 Pin WW 1 level	.70
24 Pin WW 2 level	.75
24 Pin WW 3 level	.85
40 Pin WW 3 level	1.25

EDGE CARD CONNECTORS

IMSA1 S100 WW .250" Row Spacing	$5.00
IMSA1 S100 ST .250" Row Spacing	5.00
Altair S100 WW Wire Wrap	5.50
Altair S100 ST Soldertail	5.25

D-SUBMIN CONNECTORS

RS-23225 Pin Male Part No. DB25P	$3.50
RS-23225 Pin Female Part No. DB25S	3.95
RS-232 Plastic Hood for above	.75

CRYSTALS

Microprocessors Timebases TV Games Precision Crystals

Frequency	Case	Price	Frequency	Case	Price
1.000 MHz	HC6/HC33	4.95	13.000 MHz	HC18	4.95
2.000 MHz	HC6/HC33	5.85	14.31818 MHz	HC18	4.95
2.097162MHz	HC6	6.95	18.000 MHz	HC18	4.95
4.000 MHz	HC18	4.95	20.000 MHz	HC18	4.95
5.000 MHz	HC18	4.95	32.000 MHz	HC18	4.95
6.000 MHz	HC18	4.95	100Kc	HC13	12.95
10.000 MHz	HC18	4.95	3579.45 Kc	HC18	4.95
			2.4576 MHz		5.95

★ Exar

★ FUNCTION GENERATOR KIT ★
The Function Generator Kit features sine, triangle and square wave. THD 0.5% typ., AM/FM capability
XR 2206KA Includes monolithic function generator IC, PC board, and assembly instruction manual 18.95

★ TIMING CIRCUITS ★

XR555CP	Monostable Timer	.90
XR556CP	Dual 555 Timer	1.75
XR2556CP	Dual Timing Circuit	2.95
XR2240CP	Programmable Timer	4.20
XR320P	Timer	1.40

★ COMMUNICATION CIRCUITS ★

XR1488	Quad RS232 Driver	1.45
XR1489	Quad RS232 Receiver	1.45
XR2208	Four Quadrant Multiplier	4.75

★ PHASE LOCKED LOOPS ★

XR2211CP	FSK Demod/Tone Decoder	5.95
XR567CP	Tone Decoder	1.80
XR2567CP	Tone Decoder	2.80

★ STEREO DECODERS ★

XR1310P	FM Stereo Demodulator	3.00
XR1310EP	FM Stereo Demodulator	3.00
XR1800	Stereo Demodulator	3.00

★ FUNCTION GENERATORS ★

XR2206CP	Monolithic Function Generator	5.25
XR2207CP	Current Controlled Oscillator	3.75

★ OPERATIONAL AMPLIFIERS ★

XR742CP	Zero Crossing AC Trigger	3.00
XR4558CP	Dual Hi Gain Op Amp	1.20
XR4136CP	Quad Op Amp	1.75
XR4202CP	Programmable Quad Op Amp	4.25

2708 PRIME ONLY $49.95

Build Your Own MONITOR

Now in Stock – NEW

M3000 - 100 12-inch display module	$229.00
M2000 - 155 9-inch display module	$208.00

*Add $10.00 Shipping

MOTOROLA Exorciser, MEK6800DI and MEK6800D2 Compatible Modules

9601 16 Slot mother board - Inc. Conn.	$175.00
9602 16 Slot card cage 19" rack mount	$ 72.00
9610 Utility Prototyping Board	$ 36.00
9615 4K Eprom Module (1702A)	$350.00
9620 Parallel I/O Module	$375.00
9626 8K Static Ram Module	$350.00
9650 8 Channel Duplex Ansyn I/O	$390.00

All assembled and tested. Not Kits

★ MISCELLANY

RIBBON CABLE · ELCO VARIPAK 2 H 2 Slot cardholders 4/$1.00 · LED'S · 3.5V BUZZER $.95

KEYBOARD

63 key array, terminal keyboard, unencoded with high quality switches. Black w/white legends. Special Offer: Buy now at $19.95 and we will give you a free calculator type keyboard.

TOUCHTONE KEYBOARD FOR TELEPHONE
Use with MK5085 N integrated tone dialer & 3.579 Hz crystal. Crystal available for $4.95
Only $9.95

NOW! A 19 key Number Keyboard compatible with our 63 Key Keyboard. Same manufacturer and design. Keys for 0 to 9 and 110 300 FDX/HDX, minus Auto LF Line Loc, Alt Char set and Char set lock. ONLY $11.95 FIRST TIME OFFERED (Identical to our 63 Key as pictured except 19 keys.)

★ CLOCK CHIPS

MM5311	$4.50
MM5312	4.00
MM5314	3.50
MM5316	4.50
MM5375	3.50

★ IC SOCKETS

★ RESISTORS

Carbon Resistors 1/4 Watt ±5%. Most Standard Resistance Values

★ VARIABLE RESISTORS

CORCOM EMI FILTER

CORCOM 284 2A · CORNELL DUBILIER NF21015 2×4A 250 VAC Only $4.95 · DIP SWITCH

ROTARY SWITCH

C&K V 25 Bat handle toggle switch $1.50

Circle 75 on inquiry card.

Circle 168 on inquiry card.

TTL SPECIALS

74H10 Dual 4 Input Buffer	$ .20
7490 Decade Counter	.49
74S175 Quad Flip-Flop with clear	.99
74283 4 Bit Binary Adder	.99

2½" round speaker, 8 Ohms	1.00
2¾" round speaker, 100 Ohms	1.10
Speco miniature replacement speakers from 1" to 3¼", SASE for list.	
Brand New GE Stereo Tape Amplifier Board with all components 4 Watts 12 V ac supply limited	3.50
Mono Amplifier Board 1 control	2.25
6 foot black or brown zip cord and plug	.35
RG174 50' coil	4.50
2¼" round speaker, 8 Ohms	.75
VHF Ferrite Beads	15 for 1.00
Ham & CB Slide Mounts with lock & coax connectors	8.95
2 Amp Bridge Rectifier, 200 Volt	.50
2 Amp Bridge Rectifier, 600 Volt	1.25
3 Amp Bridge Rectifier, 200 Volt	.85
Gould AA 500 mA Nicads	1.69
2 Amp 1000 volt rectifier	10 for 1.00
LM 723 14 pin Voltage Reg	.49

RF DEVICES

2N3375 3W 400 MHz	$5.50
2N3866 1W 400 MHz	1.15
2N5589 3W 175 MHz	4.75
2N5590 10W 175 MHz	7.80
2N5591 25W 175 MHz	10.95
2SC517	3.95
2SC1226	1.25
2N6080 4W 175 MHz	5.40
2N6081 15W 175 MHz	8.45
2N6082 25W 175 MHz	10.95
2N6083 30W 175 MHz	12.30
2N6084 40W 175 MHz	16.30
2SC1306	4.30
2SC1307	5.25
2N2876	special 10.95

ZENERS

1N746 to 1N759	400 Mw	ea. .25
1N4728 to 1N4764	1 watt	.28
1N5333 to 1N5378	5 watt	2.10
1N2970 to 1N3005	10 watt	2.40
1N3305 to 1N3340	50 watt	4.75

.001 Pacer Cap. 192P10292 200 WVDC	.18
12.8 kHz Crystal in TO 5 Can	4.95

8038	$3.25
MPSA14	.90
2N3055	.99
MPF102 FET	.55
2N3904 or 2N3906	.25
2N5496 or 2N6108	.70
MJE340 (2N5655)	1.10
40673 RCA FET	1.55
741 or 709 14 Pin DIP	.25
555 Timer	.75
556 Dual 555	1.75
200 Volt 25 Amp Bridge	1.50
1N914-1N4148	15 for .99
1N34-1N60-1N64	10 for .99
CA 3028 Dif. Amp	1.50
4060 CMOS	2.00
LM309K Volt Reg	1.10
MJ3055	2.20
5313 Clock Chip	3.95
5314 Clock Chip	4.50
5316 Clock Chip	4.95
2N6103	.89
LM309 or 741 Min DIP Op Amp	.45
LM741CE T05 Op Amp	.45
14 or 16 Pin IC Sockets	.30
Slide Pots Tapered 1 K or 15 K	$ .50
Egg Insulators	.45

2N918	.95
2N2218	.45
2N2219A	.40
2N2222A	.30
2N2369	.20
2N2483	.34
2N2484	.45
2N2905	.35
2N2907	.25
2N2926G	.24
2N2926Y	.24
2N3053	.50
2N3390	.75
2N3439	1.59
2N3440	.60
2N3512	1.15
2N3553	1.40
2N3565	.22
2N3584	.30
2N3638A	.37
2N3646	.27
2N3713	1.35
2N3725A	1.80
2N3771	2.50

ALDELCO COMPUTER CENTER NOW OPEN

Kits, Books, Boards, Magazines. Special 2102LI 8 for $17.50. 8080A CPU Chip $29.95. We stock OK Battery Operated Wire Tool $34.95, OK Hand Wire Wrapped Tool $5.95. 7400 ICs CMOS, Timers PLL's IC Sockets. All kinds of transistors, rectifiers and diodes. Plus other electronic parts.

HOURS: MONDAY TO SATURDAY, 9:30–5:00
OPEN WEDNESDAY UNTIL 9 PM (516) 378-4555.

We quote on any device at any quantity. Add 5% for shipping. Minimum order $6.00. Out of USA send certified check or money order, include shipping costs.

Special 50 Foot Spool #30 wire wrap $1.98. White, blue, red or yellow.

NATIONAL 8080A SPECIAL $19.95

2N3772	2.25
2N3773	4.95
2N3859	.29
2N3903	.20
2N3905	.25
2N3924	1.80
2N3926	6.30
2N4041	7.80
2N4249	.25
2N4401	.25
2N4402	.25
2N4403	.25
2N4409	.19
2N4427	1.35
2N4429	7.65
2N4888	.50
2N5016	17.60
2N5090	7.50
2N5129	.40
2N5179	.90
2N5641	5.40
2N5642	10.25
2N5643	14.35
2N5913	1.70

Build the W7BBX Programmable Keyer (Ham Radio April 1976) We can supply the four PC boards and a Comprehensive Construction Manual all for only $29.95

Write for our catalog featuring other Kits including the hard to get Keyer switching transistor 2N4888 as well as other Keyer parts and boards.

Clubs contact us for quantity discounts on any of our kits.

We have Wire Wrap Sockets and Wire Wrap Wire – 50 feet $1.98.

ALDELCO KITS

STOPWATCH KIT Operates on a 9 Volt battery. Includes Crystal, Switches, 7205 MOS Chip & LED Displays and Board $29.95
STOPWATCH HAND CASE for above 3.95
CLOCK CABINETS Beautiful wood simulated walnut grain $3.95 Plexiglass in Blue, White, Black or Smoked $2.95

SIX DIGIT AUTO OR BOAT DIGITAL CLOCK KIT Has a beautiful Charcoal Grey moulded high temperature plastic case with chrome rim. Dimensions are 1¾" high x 4" wide x 4½" deep. Large 0.4 LEDs display hours, minutes and seconds. Works on 12 Volt AC or DC as well as automatic switching to a 9 Volt battery for power failures. Battery (not supplied) fits in case. Provision for blanking display LEDs for out of car or boat use. Adjustable Crystal Time Base included, as well as Mobile Mounting Bracket.

KIT ONLY	$29.95
Three or more kits	27.95
Wired and Tested	39.95
Power Pack for use on AC	2.95

NOW NEW IMPROVED DIGITAL ALARM CLOCK KIT Hours · Minutes · Seconds displayed on six BIG 0.5 Fairchild 7 Segment Display LEDS 12-hour format 24-hour alarm with snooze feature, plus elasped time indicator and freeze feature. Eight pages of pictorials and instructions. NEW on-board power transformer and circuitry for optional time base $19.99
60 Cycle time base kit for dc use in automobile or for battery operation $4.95

12 OR 24 HOUR CLOCK KIT. Comes with Big 0.5 Seven Segment LEDs. Uses National 5314 Clock Chip. Fits our Walnut Grain or Plexiglas Cabinets. ONLY $18.95

ALDELCO

2281A BABYLON TURNPIKE, MERRICK NY 11566
516-378-4555

California Industrial

Post Office Box 3097 B • Torrance, California 90503

SPERRY UNIVAC KEYBOARD

$24.88 Univac

The famous Sperry Univac 1710 Hollerith keyboard assembly is now available from California Industrial for only $24.88 The ideal computer input device for accountants and mathematicians. The numeric keys are placed on the lower three rows to resemble a ten key adding machine. This format allows one handed numeric data entry.
Original cost was $385. Used but guaranteed in excellent condition. Complete with documentation.

CONNECTORS

RS-232

DB25P male plug & hood **$3.95**

DB25S female **$3.95**

$4.98 100 PIN IMSAI/ALTAIR Edge Connector

Altair, Imsai compatible gold plated, dual 50 (.125 centers) three tier wire wrap edge connector. 3 for $13.50

$17.50 63 key KEYBOARD

Uncoded computer keyboard. Contains: 63 reliable gold plated SPST switches. ASCII Kit: components and printed circuit board. $35.00 additional.

JOYSTICK $5.50

This joystick feature four 100K potentiometers, that vary resistance proportional to the angle of the stick. Perfect for television games, quad stereo and radio controlled aircraft.

POWER SUPPLIES

$17.50

$1.50 shipping

5 volt 2.2 Amp regulated power supply. Also delivers 12 volts at 4 Amps unregulated. Perfect for TTL applications.

Power Transformer

$4.98 $1.50 shipping

Input 117ac. Output-21v 5A center tapped. Suitable for 12v. 5A. or 5v. 10A power supply. Four other windings at lower currents.

CALCULATOR KEYBOARD

$2.98

Ideal for keyless entry systems, burglar alarms, Touch Tone or hexadecimal computer input code.

$3.98 Digital Clock

Manufactured for the Panasonic clock radio. The clock mechanism trips a microswitch upon reaching your preset wake-up time.

intel 2708

$29.95

8K UV Erasable MEMORY

SUPERECTIFIER

$2.50

BRIDGE RECTIFIER

MOTOROLA 27 Amp. 200v.

$3.50 BNC CABLE

15 feet of RG-58U connector at ends

TTL As Low As $.09*

7400	.09	7476	.39
7401	.19	7479	3.99
7402	.19	7480	.79
7403	.19	7482	.99
7403	.19	7483	.99
7404	.19	7485	.99
7405	.19	7486	.49
7406	.19	7488	3.40
7407	.25	7489	2.79
7408	.25	7490	.49
7409	.25	7491	.99
7410	.19	7492	.49
7411	.25	7493	.49
7412	.35	7494	.79
7413	.49	7495	.79
7414	.79	7496	.79
7416	.39	7497	3.99
7417	.39	74100	1.19
7420	.19	74107	.39
7422	.49	74109	.49
7423	.39	74110	1.79
7425	.39	74116	1.99
7426	.39	74120	1.79
7427	.39	74121	.39
7428	.49	74122	.39
7429	.39	74123	.69
7430	.25	74125	.59
7433	.39	74126	.59
7437	.39	74128	.49
7438	.39	74132	.99
7439	.39	74136	.89
7440	.39	74141	.99
7441	.89	74145	.99
7442	.59	74147	2.49
7443	.79	74148	1.99
7445	.89	74150	1.19
7446	.99	74151	.99
7447	.99	74153	.89
7448	.99	74154	.99
7450	.25	74155	.99
7451	.25	74156	1.29
7453	.25	74157	.99
7454	.25	74159	2.99
7460	.25	74160	1.19
7470	.39	74161	.99
7472	.39	74162	1.49
7473	.39	74163	.99
7474	.35		
7475	.49		

74164	1.19	4049	.79
74165	1.19	4050	.79
74166	1.19	4051	1.99
74167	4.99	4052	1.49
74170	2.49	4053	2.19
74173	1.49	4060	3.19
74174	1.19	4066	1.19
74175	.99	4069	.49
74176	.99	4071	.49
74177	.99	4077	.79
74180	.99	4081	.49
74181	2.49		
74182	.99		
74184	1.99		
74185	1.99		
74186	4.99		
74190	1.19		
74191	1.19		
74192	.99		
74193	.99		
74194	1.19		
74195	.99		
74196	1.19		
74197	.99		
74198	1.49		
74199	1.99		
74273	1.99		
74290	.99		
74366	.99		
74390	1.79		
74939	1.79		
74490	1.99		

MEMORY

1702	9.95
82s23	2.95
82s123	2.95
2102	1.79
2102-1	1.89
21L02	1.99

CPU's

8080A	24.95
6800	24.95
Z80	69.95

CLOCK's

5314	3.95
5316	4.95
5375	3.95

LM741 $.09*

CMOS

4000	.25
4001	.25
4002	.25
4006	1.99
4007	.25
4008	1.49
4009	.69
4010	.69
4011	.25
4012	.25
4013	.49
4014	1.49
4015	1.39
4016	.69
4017	1.29
4018	1.69
4019	1.79
4020	1.39
4021	1.49
4022	1.25
4023	.25
4024	1.19
4025	.25
4027	.69
4028	1.25
4029	1.99
4030	.69
4032	.49
4033	1.99
4035	1.69
4040	1.99
4041	1.49
4042	1.49
4043	1.49
4044	1.49
4046	2.49
4047	2.49

300H	.79	340T-24	1.79
301H	.39	350N	.99
301CN	.39	351CN	.65
302H	1.29	370H	1.29
304H	1.29	370N	1.29
305H	.99	373N	3.19
307H	.49	377N	3.99
307CN	.99	380N	1.39
308H	.99	381N	1.79
308CN	.99	382N	1.79
309H	1.09	NE555v	.49
309K	.99	NE556	.99
310H	1.19	NE565H	1.49
310CN	1.19	NE565N	1.49
311H	.99	NE566N	1.25
311N	.99	703CN	.45
312H	1.79	709H	.39
318H	1.49	709N	.39
318CN	1.29	710N	.79
319CN	1.39	711H	.39
320K-5	1.39	711N	.39
320K-12	1.39	723H	.55
320K-15	1.39	723N	.55
320T-5	1.75	725H	3.49
320T-8	1.75	733H	1.49
320T-12	1.75	733N	.99
320T-18	1.75	739N	1.19
320T-24	1.75	741N	.09
324N	1.79	747N	.79
339N	1.69	748H	.39
340K-5	1.95	748N	.39
340K-8	1.95	1414N	1.75
340K-12	1.95	1458	.69
340K-15	1.95	1496N	.95
340K-18	1.95	2111N	1.95
340K-24	1.95	2901N	2.95
340T-5	1.79	3065N	.69
340T-12	1.79	3900N	.49
340T-15	1.79	3909N	1.19

*We want to give them away — but our accountants won't let us. So instead we are selling them at cost. And therefore must limit the purchase of the SN7400 and the LM741 to 50 per customer.

Compucorp DIGITAL CASSETTE RECORDER

$79.50

The Compucorp 392 cassette recorder is engineered exclusively for storage and retrieval of binary digital information. The recorder does not require tone detection or analog interface boards. High-low binary state is detected through a self contained digital transition amplifier. The upper half of the recorders stereo head is used for detection of self clocked pulse signals. Information received from the CPU advances the capstan driven tape transport. Control cable terminates into a 25 pin male "RS-232" type connector. Documentation included. Limited quantities.

SOLID-STATE MICRO BUZZER

$.98

1.5 TO 3 VDC

TRIMMER POTENTIOMETERS

2K 10K 50K

5 for $.98

20	50	100
16¢	14¢	12¢

Thumbwheel switch

Ten position BCD

$1.39 ea.

10	50
$1.19	.89

SPDT MINIATURE TOGGLE SWITCH

$.98 ea.

10	50	100
$.88	.81	.73

CAPACITORS

ELECTROLYTICS

	ea.	10	50
4500/50v.	$1.49	1.35	1.19
1000/15v. axial	$.55	.49	.45
.1 disc	$.12	.10	.08
.01 disc	.06	.05	.04

DPDT ROCKER SWITCH

$.98 ea.

Photoconductor Cells

3 for $.98

10 ohm to 100 K

32 Conductor RIBBON WIRE

$.49 Ft.

SPECTRA-STRIP

COMPUCORP Power Adapter

shipping $.75

output 7vdc 1.4A

$4.98

COMBINATION LENS MOUNTING DEVICE

5 for $.98 Red Amber Green

20	50	100	1k
17¢	15¢	13¢	11¢

CLIP LITE

Transistors

	ea.	10	50	100
2N2222A	.20	.18	.16	.15
2N3055	.89	.84	.77	.65
MJ3055	.99	.94	.87	.75
2N3904	.15	.11	.09	.07
2N3906	.15	.11	.09	.07

Diodes

		10	25	100
1N4002	100v.	.08	.06	.05
1N4005	600v.	.10	.08	.07
1N4148	signal	.07	.05	.04

jumbo red LED's

ea.	10	25	100
$.15	.10	.08	.07

TRANSFORMER

$1.49

Output: 12v. 175ma.

5-WAY BINDING POSTS

3 For $.98

9 foot

$1.49

Heavy duty grounded power cord and mating chassis connectors.

Nicad

$1.19

AA NICAD BATTERY

DISCOUNT Wire Wrap Center

IC SOCKETS

pin	Wire Wrap ea.	25	50	Solder ea.	25	50
8				17¢	16	15
14	37¢	36	35	20	19	18
16	38	37	36	21	20	19
24	99	93	85	36	35	34
40	169	155	139	63	60	58

50ft. $.98 KYNAR WIRE WRAP

500	1,000	11,000
$9.	$15.	$105.

Battery wire wrapping tool ~~$34.95~~ $29.95

OK HOBBY WRAP-30 wire wrap & strip tool $5.45

15214 Grevillea Avenue • Lawndale, California 90260 • (213) 772-0800

All merchandise sold by California Industrial is premium grade.
Orders are shipped the same day received.
PLEASE INCLUDE $1.00 SHIPPING ON ORDERS UNDER $15.00
California residents add 6% sales tax • Money back guarantee.
Sorry, no COD's • Foreign orders add 10%

Circle 200 on inquiry card.

COMPUTER WAREHOUSE STORE


DEPT: B • 584 COMMONWEALTH AVENUE • BOSTON, MA • 02215 • 617-261-2701 • VISIT US: 9-9 WEEKDAYS; 9-6 SATURDAYS

ONE DAY SHIPMENT

USED PERIPHERALS FOR MICROSYSTEMS

HAZELTINE 1000 $795

VIDEO DISPLAY TERMINAL + SHIPPING 35 lb.
12 LINES x 80 CHAR., TTY COMPATIBLE,
5 x 7 DOT MATRIX, 525 LINE RASTER.
BUILT & TESTED; PLUG & GO

ALL an ASR 33 is and MORE!

OLIVETTI TE318 - BACK IN STOCK!
RS232 INTERFACE, QUIET OPERATION
10 CPS, PRINTER, BUILT-IN PAPER TAPE READER/PUNCH, ELECTRIC TYPEWRITER KEYBOARD WITH ADDITIONAL 10 KEY NUMERIC PAD, YOUR CHOICE OF FRICTION OR SPROCKET FEED, LIGHTED PLATEN AREA FOR EASY READING, STANDARD PAPER AND TAPE, SUPPORTED BY OLIVETTI

$875 + SHIPPING 165 lb

TECHTRAN 4100.......$595 + $25 SHIPPING

TAPE CASSETTE DRIVE. CAN RUN DIRECTLY FROM TERMINAL INDEPENDENT OF CPU. FULL EDIT CAPABILITY.

KDI ADTROL AR-21...$95 + $25 SHIPPING

ELECTRO OPTICAL PAPER TAPE READER WITH 110V PS, STEPPER MOTOR, 250 CHAR/SEC, FAN IN TABLE TOP HOUSING

IBM 731 I/O WRITER $750 + $25 SHIPPING

8½"PLATEN,PINFD,EBCDIC,U/L CASE,DUAL CLR RIBBON,115V

CWS U-BUILD-IT SYSTEM $599

+ $35 SHIPPING
MPU, CRT TERMINAL & AUDIO CASSETTE AT A ROCK BOTTOM PRICE !

SC/MP MICROCOMPUTER; 256 x 8 RAM, 512 x 8 ROM
GREEN PHOSPHOR VIDEO MONITOR; 12" RASTER SCAN
SWTPC 1024; 32 CHAR. x 16 LNS, ASCII KEYBOARD
SWTPC AC-30 AUDIO CASSETTE INTERFACE
*OTHER PACKAGE PLANS - SEND FOR OUR CATALOG

TO ORDER EQUIPMENT

1. ENCLOSE CHECK FOR FULL PRICE PLUS SHIPPING CHARGES (KITS - ADD $5 IF UNDER $100; $10 IF OVER) (FOREIGN RATES HIGHER)
   BANKAMERICARD & MASTERCHARGE ACCEPTED -
   SEND CARD NUMBER, EXPIRATION DATE, INTERBANK #
2. CLEARLY IDENTIFY SHIPPING ADDRESS
3. DESCRIBE ITEM BY MODEL NUMBER

ALL MERCHANDISE WARRANTEED

A MUST FOR PERSONAL COMPUTING

SEND $1 FOR OUR CATALOG

THE ONLY ONE OF ITS KIND! FULL DETAILS ON OUR COMPLETE LINE OF KITS & UNITS, REVIEWS OF OVER 150 BOOKS, LISTS OF NEW & SURPLUS PARTS & "ALL ABOUT HOBBY MICROCOMPUTERS" - AN INTRODUCTION TO PERSONAL COMPUTING.

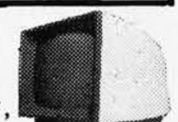
Circle 138 on inquiry card.

SPECIAL DISCOUNTS! ON KITS & ASSEMBLED UNITS

SAVE UP TO 20% OFF KIT PRICE WHEN A PERIPHERAL IS PURCHASED AT THE SAME TIME. ($200 MAXIMUM DISCOUNT)

PERIPHERAL PRICE OVER $900 ➡ 20% OFF KIT
PERIPHERAL PRICE OVER $250 ➡ 10% OFF KIT
PERIPHERAL PRICE OVER $95 ➡ 5% OFF KIT

GREEN PHOSPHOR $150 VIDEO MONITOR +25 SHIPPING



STANDARD 1V P TO P COMPOSITE VIDEO INPUT, 16MHz BAND WIDTH, RASTER SCAN 12x12x13" WITH POWER SUPPLY, VIDEO AMPLIFIER, DRIVING CIRCUITRY VENTILATION MUFFIN FANS, 7x9" HORIZONTAL VIEWING CAPABLE OF 24 LINES x 80 CHAR., ANTIGLARE ¼" ETCHED GRADIENT DENSITY FACE PLATE, P39 GREEN PHOSPHOR, ON/OFF BRIGHTNESS CONTROLS, 115Vac, 60 W ... TRULY A COMMERCIAL UNIT BUILT TO WORK IN A DEMANDING ENVIRONMENT.

DATAPOINT 3300-200

THERMAL PRINTER **$375** +$25 SHIPPING
PARALLEL PRINTER WITH ADDITIONAL CIRCUIT BOARDS TO PROVIDE SERIAL INTERFACE, PRINTS UP TO 30 CPS, 100Vac PS USES WIDELY AVAILABLE NCR PAPER, 96 CH, ASCII, 80 COL, CRT COMPATIBLE 5x7 DOT MATRIX, SOLID STATE WITH LESS THAN 25 MOVING PARTS.

DATAPOINT 2200-200 $395 +285 lb SHIPPING

CONSOLE PRINTER
BOTH UNIVAC & SINGER BUILT THESE PRINTER MECHANISMS WHICH OPERATE AT 30 CPS FROM A ROTATING WHEEL. 65 CHAR. USES STANDARD FORM FEED PAPER, PINWHEEL IS INTERCHANGEABLE.
UNIVAE 0769-06 PRINTER MECHANISM ONLY INCLUDES MOTOR/PRINT WHEEL ... $195 +SHIPPING 75 lb.

KITS ★ HIGHLIGHTS FROM OUR WIDE SELECTION ★

IMSAI 8080 MICROKITS

8080A KIT 5 SLOT................$699
8080A KIT 22 SLOT............... 751
4K MEMORY KIT................... 139
PIC-8 PRIORITY INTERRUPT........ 125
SERIAL I/O KIT.................. 125
PROM 4-512 KIT.................. 165
UCRI-1 KIT...................... 59
CABLE A KIT..................... 18
STANDARD INPUT/OUT INTERFACES, MANUALS, SOCKET SETS

VIKING 100 PIN, HEAVY DUTY.....$3.00

SWTPC 6800..........$395

•512 BYTES of ROM •SERIAL INTERFACE
•RS232 or 20 mA •4K of RAM

4K MEMORY.......................$100
MPA...........145 MPB.......... 40
MPC........... 40 MPD.......... 35
MPE.........14.95 MPF.......... 30
MPM........... 65 MPMx......... 35
MPP........... 35 MPL.......... 35
MPS........... 35 MPAb........14.50
MPMb........14.50
MPCb, MPSb, MPLb, EACH..........9.50
CONNECTOR SETS MPU/MEM..........2.50
CONNECTOR SETS INTERFACE........ 2
4KBA.......... 5 GT61......... 99
AC30 AUDIO INTERFACE...........79.50
PP40 PRINTER................... 250
CT 1024 TERMINAL KIT........... 275
CTP.........15.50 CTS.........39.95
ALL SWTPC UNITS ARE KITS

SMOKE SIGNAL BROADCAST 16K RAM 595

LEAR SIEGLER ADM-3A W/ Cursor Control

•12" CRT
•24 LN X 80 CHAR
•RS232
•20 mA LOOP **$875**

SCAMP KIT...............$99

FROM NATIONAL SEMICONDUCTOR
KEYBOARD KIT........... 95

★ BUILT UNITS ★

ICOM MICROFLOPPIES

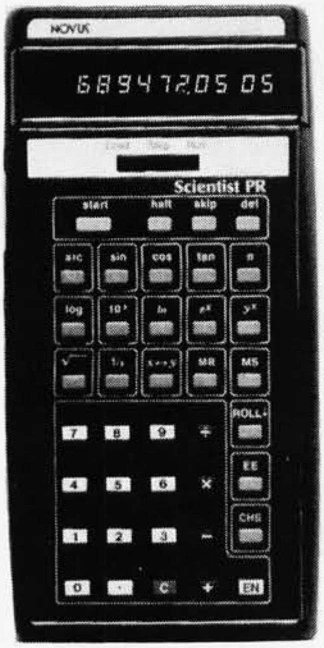
PLUG COMPATIBLE FOR:
S100 BUS...FD2411..... **$1095**
SINGLE DRIVE FD2402......$ 649
ICOM FLOPPIES:
FF36-1 FRUGAL..........$1195
FF36-2 DUAL FRUGAL..... 1095
360-58 BLT;INTFC 8080.. 300
S171 POWER SUPPLY ... 250
FD360-2-5DUAL SYSTEM... 3000

KIM-1 6502...............$ 245

KIM-2 4K................ 179
KIM-3 8K................ 289
MANUALS PACKAGE.......... 15
TARBELL AUDIO CASSETTE KIT 120

INTERSIL INTERCEPT JR. $281

12K RAM................ 145
ROM/PROM BOARD.........74.65
YOU ADD MEMORY CHIPS
SERIAL I/O.............81.50
AUDIO VISUAL BOARD..... 125



SPECIAL OFFER 100 STEPS PROGRAMMABLE SCIENTIST

#4525

REG. $89.95

$38.95

· RPN logic with "built-in" hierarchy for increased accuracy and speed in calculating sequences involving arithmetic, trigonometric, logarithmic, power or exponential functions. · A rollable 4-level stack lets you review or use intermediate solutions. · Eight-digit plus 2-digit exponent LED display with full-floating decimal system. · Scientific notation for increased mathematical capacity. · Sine, cosine, tangent and inverse trigonometric functions. · Common and natural logarithms and antilogarithms. · Instant automatic calculation of powers and roots. · Single-key square root calculations. · Single-key Pi entry. · Separate storage memory. Square, square root and reciprocal calculations. Change sign and register exchange keys. · MOS/LSI solid-state circuitry. · Includes 3 AA rechargable NiCad batteries.

MANUFACTURED IN USA BY
BY NATIONAL SEMICONDUCTOR
ONE YEAR UNLIMITED WARRANTY
10 DAYS MONEY BACK GUARANTEE

____ Model #4525 @ 38.95 ____
____ Model #4520 @ 28.88 ____
____ AC Charger @ 4.95 ____
____ Case(s) @ 2.95 ____
____ Calc. Stand @ 2.50 ____
Cal. Res. add 6% tax ____

Total ____

ALSO AVAILABLE

MODEL #4520 With the same features as above but without programming capability @ $28.88. Accessories same price as above. Circle 157 on inquiry card.

SAME DAY SERVICE IF PAID BY CASHIER'S CHECK, MONEY ORDER OR CHARGE CARD.

ASC II KEYBOARD
NEW LOOK IMPROVED DESIGN

$58.00

This 63 key **ASC II Encoded Keyboard** kit was designed and manufactured by **Electronics Warehouse Inc.** Features: Single 5 volt D.C. supply, utilizing only TTL Logic elements (no MOS devices to blow). TTL drive capability (each of the eight bits of ASC II output will drive the equivalent of ten standard TTL inputs without external buffer drivers), de-bouncing, upper and lower case fully ASC II, 8 bit parallel. In addition to the alpha-numeric and symbol keys available on a regular keyboard, the following keys are utilized: Escape, back-space, tab, line-feed, delete, control, shift-lock, shift (2 keys), return. All 128 ASC II characters are generated.

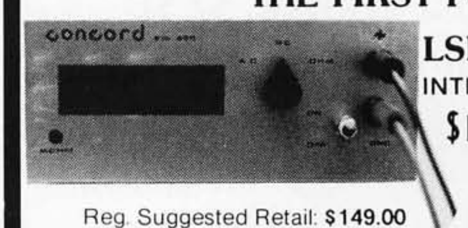
Kit includes: 63 key keyboard, P.C. board, all required components and assembly manual with ASC II code list.

Optional: Parity bit - add 50¢ • Enclosure - $25.00 • Serial output - add $2.00 • 18 Pin edge connector - $2.00 • Sockets - $4.00

Note: If you already have this teletype keyboard you can have the kit without it for **$39.00.** Dealer inquiries invited.

FROM CONCORD

THE FIRST FULL *FEATURE* LSI DMM KIT



INTRODUCTORY PRICE:

$77.77

Reg. Suggested Retail: $149.00

- AUTO RANGING
- AUTO POLARITY
- AUTO ZERO
- 3 Large Digits (1/2")
- Rechargable

MEASUREMENT RANGES:

Voltage: (AC & DC) 1 MV - 1000V
Current: (AC & DC) 10 µA - 1A
RESISTANCE: 1 Ω - 10 MΩ
Basic D.C. Accuracy, better than 0.1% ± 1 Digit
Power: 4 AA batteries (Rechargable batteries optional)

NI-CAD BATTERIES: $6.00 • AC CHARGER: $4.95 • ENCLOSURE: $12.95 • TEST LEADS: $1.95 • SHUNT KIT FOR 3 CURRENT RANGES: $4.75 • SOCKETS $2.50

ORDERING INFORMATION

SHIPPING AND HANDLING - $3.00 + 50¢ Insurance
California residents add 6% sales tax

ELECTRONICS WAREHOUSE Inc.

1603 AVIATION BLVD. Dept. B
REDONDO BEACH, CA. 90278
TEL. (213) 376-8005

WRITE FOR FREE CATALOG

You are invited to visit our store at the above address

A New Generation of Computer Kits

When we started catering to the computer hobbyist back in '73, some people thought we were going to go out of business; now it's 1977, and because of the popularity of our computer kits we are again expanding the line. Note particularly the new 8K ECONORAM IItm: first the price, which is extremely low; next the unique configuration that extends its usefulness. Our Motherboard represents another tremendous value made possible because of our large parts business. We're also distributing products from both George Morrow and Mullen Computer Boards---we think they are exceptionally good and represent cost effectiveness equal to our line of kits.

From parts to peripherals, we are your one-stop, mail order computer store.

8Kx8 ECONORAM II T.M.

Uniquely configured as 2 separate 4K x 8 blocks (with fully independent protect and address decoding to increase flexibility), our ECONORAM II™ features full buffering, guaranteed 450 ns or faster speed (use 1 wait state with Z-80; necessary logic included on board), plated-through, double-sided epoxy glass board with gold edge fingers, low profile sockets, on board regulation, and 3 state outputs that can drive the Altair/IMSAI S-100 buss or any bidirectional buss.

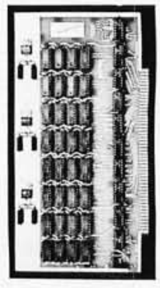
We use proven, reliable technology, like static 2102L-1 low power 1K RAMs, and low power Schottky ICs which keep current consumption to an absolute minimum. And there's more...to see for yourself, send $1 to cover costs and we'll send you our ECONORAM II™ Logic Print/Documention.

$163.84
¼¢ per bit!

4Kx8 Econoram T.M.

If you want the best combination of value and economy, look at ECONORAM™. We don't just claim low power: we guarantee current consumption under 750 mA, with the typical board falling between 600 and 650 mA. Fully compatible with the S-100 buss; full buffering gives clean and unambiguous data transfer; and speed is guaranteed 450 ns or faster. We didn't cut corners on the kit either, with low profile sockets, dipswitch address selector, low power Schottky support ICs, and a high quality circuit board.

Also available assembled, tested, and warranted for one year for **$130.** **$100**



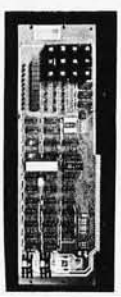
MOTHERBOARD $80

Use with the IMSAI microcomputer as an add-on with room for 10 peripherals, or for starting an 11 slot stand-alone system. Comes with 10 edge connectors---lots of places would charge you the cost of our Motherboard for these alone. Active, regulated terminations, included on board, minimize the crosstalk, noise, overshoot, and ringing you can find on improperly terminated busses. And of course, we use an epoxy glass board, with bypass caps and heavy power traces included. Get yourself one of these, a Morrow Sigma-100 Front Panel, some ECONORAM II boards and a power supply, and you're well on your way to a powerful system at a really good price.

Small System Power Supply

Finally, a quality, cost-effective supply for small systems. Gives you a full 4 Amps at 5 Volts, with crowbar overvoltage protection, along with half an Amp of +12 and half an Amp of -12...and an additional 10 mA supply, adjustable over 5 to 10V for biasing required by some CPUs. Although intended to be used with computer systems, it's also a dandy little bench supply for digital experiments. **$45**

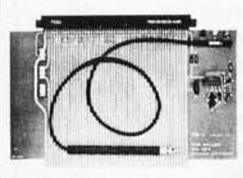
Morrow Sigma100 Front Panel



At last, a machine where you can examine, alter, and monitor every function of the CPU/front panel and its operation in real time. Edit or modify your program while you run it...think of what this means in terms of extra productivity, reduced frustration, and greater speed.

The SIGMA - 100 Minicomputer CPU Board (with integral front panel) gives you this control...run your program, or step it at any rate from 1 to 1000 steps per minute. You can stop the machine to examine and alter processor registers, memory locations, and I/O devices---there is special firmware to keep the CPU from going to sleep. You can also monitor all of the above during execution of a program as well. Everything is front panel controlled by your fingertips through a 12 pad keyboard; octal data reads out on 7 segment readouts. DOCUMENTATION: $5. **$250**

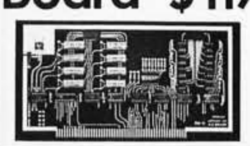
Mullen Extender Board $35



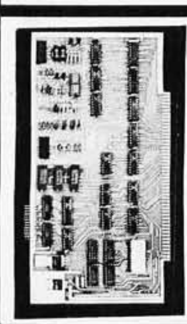
Almost every S-100 buss computer owner will at some point need an extender board, and there are many good reasons to choose this one. Like the integral logic probe... the specially designed edge connector that allows use of clip lead probing and identifies pin numbers...the jumper links that allow easy current measurement...the nonskid probe...the quality board ...and the instructions, which are clear and very complete. Save yourself troubleshooting time and trouble with this useful peripheral.

Mullen Opto-isolator/Relay Board $117

8 fast reed relays respond to an 8 bit word: feed the relay associated with its bit a "1" and it closes, give it a "0" and it opens. Also, 8 opto-isolators accept an 8 bit word from the outside world and send it to your computer for handshaking or further control purposes. Use it when you need a general purpose I/O switching gizmo - model railroad, security systems, audio switching, and so on.



Morrow I/O Board



If you think I/O boards don't offer enough features...we agree. This one handles 3 cassette machines, a modem or teletype, and video terminal or other RS-232 device...and includes a general purpose 8 bit port for an ASCII keyboard, tape reader, or the like. Contains ½ Kbyte of onboard RAM and ½ Kbyte of onboard ROM to give this peripheral the required smarts; data buffers handle the interface between the I/O board and your S-100 buss micro or minicomputer. Want to know more of this versatile board? Our complete documentation, many pages of useful information, is available for $5. **$120**

Vector 8800V Prototype Board $19.95

Same size as typical Altair card, but useful for any prototyping work. Accepts virtually any size IC package, has a power and ground plane, room for 4 regulators (with 1 heat sink provided)...implement your own memory boards, I/O boards, and other circuits.

PARTS

FND359 .4" READOUT

Common cathode, 7 segment display.
FIFTY CENTS EACH!
10/ $ 4.00
100/$35.00
actual size

FND503/510 .5" READOUTS

FND510 Com anode
FND503 Com cathode
95¢ each
10/$8.50
actual size

100 PIN EDGE CONNECTORS

S-100-140ST: NEW! Soldertail type, fits .140" spacing--same as Altair motherboards.	$6 each	5/$27.50
S-100WW: 3 level, gold plated wrap posts; .250" pin spacing fits IMSAI motherboards.	$5 each	5/$22.00
S-100ST: Same as above but soldertail.	-same-	

CAVE graphix

TERMS: Add 50¢ to orders under $10. Please allow 5% for shipping; any excess refunded. Street address MUST be included for COD orders. Place BankAmericard® and Mastercharge® orders ($15 minimum) by calling (415) 562 - 0636, 24 hours. California residents please add sales tax.

GODBOUT
BILL GODBOUT ELECTRONICS
BOX 2355, OAKLAND AIRPORT, CA 94614

Circle 9 on inquiry card.

FREE FLYER: Give us the word and we'll send you a copy of our flyer, which describes our complete line of products in greater detail. In addition to our computer oriented line, we carry many items relating to electronic music as well as a complete line of semiconductors and components.

7400N TTL

SN7400N	.16	SN7459A	.25		
SN7401N	.16	SN7460N	.22		
SN7402N	.21	SN7470N	.45	SN74154N	1.00
SN7403N	.16	SN7472N	.39	SN74155N	.99
SN7404N	.18	SN7473N	.37	SN74156N	.99
SN7405N	.24	SN7474N	.32	SN74157N	.99
SN7406N	.20	SN7475N	.50	SN74160N	1.25
SN7407N	.29	SN7476N	.32	SN74161N	.99
SN7408N	.25	SN7479N	5.00	SN74163N	.99
SN7409N	.25	SN7480N	.50	SN74164N	1.10
SN7410N	.18	SN7482N	.98	SN74165N	1.10
SN7411N	.30	SN7483N	.70	SN74166N	1.25
SN7412N	.33	SN7485N	.89	SN74167N	5.50
SN7413N	.45	SN7486N	.39	SN74170N	2.10
SN7414N	.70	SN7488N	3.50	SN74172N	8.95
SN7416N	.35	SN7489N	2.49	SN74173N	1.50
SN7417N	.35	SN7490N	.45	SN74174N	1.25
SN7420N	.21	SN7491N	.75	SN74175N	.99
SN7421N	.33	SN7492N	.49	SN74176N	.90
SN7422N	.49	SN7493N	.49	SN74177N	.90
SN7423N	.37	SN7494N	.79	SN74180N	.99
SN7425N	.29	SN7495N	.79	SN74181N	2.49
SN7426N	.29	SN7496N	.89	SN74182N	.95
SN7427N	.37	SN7497N	4.00	SN74184N	1.95
SN7429N	.42	SN74100N	1.00	SN74185N	2.20
SN7430N	.26	SN74107N	.39	SN74186N	15.00
SN7432N	.31	SN74121N	.39	SN74187N	6.00
SN7437N	.27	SN74122N	.39	SN74188N	3.95
SN7438N	.27	SN74123N	.50	SN74190N	1.19
SN7439N	.25	SN74125N	.60	SN74191N	1.25
SN7440N	.15	SN74126N	.60	SN74192N	.89
SN7441N	.89	SN74132N	1.09	SN74193N	.89
SN7442N	.59	SN74136N	.95	SN74194N	1.25
SN7443N	.75	SN74141N	1.15	SN74195N	.75
SN7444N	.75	SN74142N	4.00	SN74196N	1.25
SN7445N	.75	SN74143N	4.50	SN74197N	.75
SN7446N	.81	SN74144N	4.50	SN74198N	1.75
SN7447N	.69	SN74145N	1.15	SN74199N	1.75
SN7448N	.79	SN74147N	2.35	SN74200N	5.59
SN7450N	.26	SN74148N	2.00	SN74279N	.90
SN7451N	.27	SN74150N	1.00	SN74251N	1.79
SN7453N	.27	SN74151N	.79	SN74284N	6.00
SN7454N	.20	SN74153N	.89	SN74285N	6.00
				SN74367N	.75

MANY OTHERS AVAILABLE ON REQUEST
20% Discount for 100 Combined 7400's

CMOS

CD4000	.25			74C04N	.75
CD4001	.25			74C10N	.65
CD4002	.25	CD4035	1.85	74C20N	.65
CD4006	2.50	CD4040	2.45	74C30N	.65
CD4007	.25	CD4042	1.90	74C42N	2.15
CD4009	.59	CD4044	1.50	74C73N	1.50
CD4010	.59	CD4046	2.51	74C74	1.15
CD4011	.25	CD4047	2.75	74C90N	3.00
CD4012	.25	CD4049	.79	74C95N	2.00
CD4013	.47	CD4050	.79	74C107N	1.25
CD4016	.56	CD4051	2.95	74C151	2.90
CD4017	1.35	CD4053	2.95	74C154	4.00
CD4019	.55	CD4060	3.25	74C157	2.15
CD4020	1.49	CD4066	1.75	74C160	3.25
CD4022	1.25	CD4069	.45	74C161	3.25
CD4023	.25	CD4071	.45	74C163	3.00
CD4024	1.50	CD4081	.45	74C164	3.25
CD4025	.25	CD4511	2.50	74C173	2.60
CD4027	.69	CD4518	2.50	74C193	2.75
CD4028	1.65	MC14566	3.00	74C195	2.75
CD4029	2.90	74C00N	.39	MC4044	4.50
CD4030	.65	74C02N	.55	MC14016	.56

LINEAR

LM300H	.80			LM1351N	1.65
LM301H	.35			LM1414N	1.75
LM301CN	.35	78MG	1.75	LM1458C	.65
LM302H	.75	LM370N	1.15	LM1496N	.95
LM304H	1.00	LM373N	3.25	LM1556V	1.85
LM305H	.95	LM377N	4.00	LM2111N	1.95
LM307CN	.35	LM380N	1.39	LM2901N	2.95
LM308H	1.00	LM380CN	1.05	LM3065N	.69
LM308CN	1.00	LM381N	1.79	LM3900N	.55
LM309H	1.10	LM382N	1.79	LM3905N	.60
LM309K	.99	NE501K	8.00	LM3909	1.25
LM310CN	1.15	NE510A	6.00	LM5556N	1.85
LM311H	.90	NE531H	3.00	MC5558V	1.00
LM311N	.90	NE536T	6.00	LM7525N	.90
LM318CN	1.50	NE540L	6.00	LM7535N	1.25
LM319N	1.30	NE550N	.79	8038B	4.95
LM320K-5	1.35	NE555V	.39	LM75450	.49
LM320K-5.2	1.35	NE560B	5.00	75451CN	.39
LM320K-12	1.35	NE561B	5.00	75452CN	.39
LM320K-15	1.35	NE562B	5.00	75453CN	.39
LM320T-5	1.75	NE565H	1.25	75454CN	.39
LM320T-5.2	1.75	NE565N	1.75	75491CN	.79
LM320T-8	1.75	NE566CN	1.25	75492CN	.89
LM320T-12	1.75	NE567H	1.95	75494CN	.89
LM320T-15	1.75	NE567V	1.50	RCA LINEAR	
LM320T-18	1.75	LM703CN	.45	CA3013	2.15
LM320T-24	1.75	LM709H	.29	CA3023	2.56
LM323K-5	9.95	LM709N	.29	CA3035	2.48
LM324N	1.80	LM710N	.79	CA3039	1.35
LM339N	1.70	LM711N	.39	CA3046	1.30
LM340K-5	1.95	LM723H	.55	CA3059	3.25
LM340K-6	1.95	LM723H	.55	CA3060	3.25
LM340K-8	1.95	LM733N	1.00	CA3080	.85
LM340K-12	1.95	LM739N	1.00	CA3081	2.00
LM340K-15	1.95	LM741CH	.35	CA3082	2.00
LM340K-18	1.95	LM741CN	.35	CA3083	1.60
LM340K-24	1.95	LM741-14N	.39	CA3086	.85
LM340T-5	1.75	LM747H	.79	CA3089	3.75
LM340T-6	1.75	LM747N	.79	CA3091	10.20
LM340T-8	1.75	LM748H	.39	CA3102	2.95
LM340T-12	1.75	LM748N	.39	CA3123	2.15
LM340T-15	1.75	LM1303N	.90	CA3130	1.39
LM340T-18	1.75	LM1304N	1.19	CA3140	1.25
LM340T-24	1.75	LM1305N	1.40	CA3600	1.75
LM350N	1.00	LM1307N	.85	RC4194	5.95
LM351CN	.65	LM1310N	2.95	RC4195	3.25

74LS00 TTL

74LS00	.29			74LS139	1.95
74LS02	.29			74LS151	1.55
74LS03	.29	74LS74	.49	74LS153	1.89
74LS04	.35	74LS75	.69	74LS157	1.55
74LS05	.35	74LS76	.49	74LS162	2.25
74LS08	.29	74LS83	1.75	74LS163	2.25
74LS10	.29	74LS85	2.49	74LS164	1.95
74LS13	.69	74LS86	.49	74LS175	1.95
74LS14	1.75	74LS90	1.25	74LS181	3.69
74LS20	.29	74LS92	1.25	74LS190	2.85
74LS26	.39	74LS93	1.25	74LS191	2.85
74LS27	.39	74LS95	1.95	74LS192	2.85
74LS28	.39	74LS96	1.89	74LS193	2.85
74LS30	.29	74LS107	.59	74LS194	1.89
74LS32	.39	74LS109	.59	74LS195	1.89
74LS40	.39	74LS112	.59	74LS257	1.75
74LS51	.29	74LS132	1.25	74LS260	.55
74LS55	.29	74LS136	.59	74LS279	.79
74LS73	.49	74LS138	1.89	74LS670	3.95

CLOCK CHIPS

MM5309	6 Digit, BCD Outputs, Reset PIN	$9.95
MM5311	6 Digit, BCD Outputs, 12 or 24 Hour	4.95
MM5312	4 Digit, BCD Outputs, 1 PPS Output	4.95
MM5314	6 Digit, 12 or 24 Hour, 50 or 60 Hz	4.95
MM5316	4 Digit, Alarm, 1 PPS Output	6.95
MM5318	Video Clock Chip, For Use With (MM5841 - $9.95)	9.95
CT7001	6 Digit, Calendar, Alarm, 12 or 24 Hour	5.95

DATA HANDBOOKS

7400 Pi... & Description of 5400/7400 ICS $2.95
CMO... ...-out & Description of 4000 Series ICS $2.95
...ar Pin-out & Functional Description $2.95
ALL THREE HANDBOOKS $6.95

Timeband by FAIRCHILD — Watches —

T201 Black Bracelet $19.95
LED

T237 White w/bracelet $29.95
T236 Yellow w/bracelet $34.95
LED

Men's & Ladies

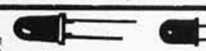
- Solid State
- Displays hour, minute, second, month & day
- Snap-out battery replacement
- Free set of replacement batteries
- Choose LED or LCD styles
- One year factory warranty

TC441 White w/strap $29.95
TC440 Yellow w/strap $34.95
LCD

LADIES LED

T311 White w/strap $34.95
T310 Yellow w/strap $39.95

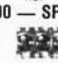
DISCRETE LEDS



.125" dia.									.190" dia.		
XC209	Red	10/$1							XC111	Red	10/$1
XC209	Green	4/$1							XC111	Green	4/$1
XC209	Orange	4/$1							XC111	Yellow	4/$1
XC209	Yellow	4/$1							XC111	Orange	4/$1
.200" dia			.185" dia.			.200" dia.			.085" dia.		
XC22	Red	10/$1	XC526	Red	5/$1	XC556	Red	10/$1	MV50 - Red - 6/$1		
XC22	Green	4/$1	XC526	Green	4/$1	XC556	Green	4/$1			
XC22	Yellow	4/$1	XC526	Yellow	4/$1	XC556	Yellow	4/$1	INFRA-RED LED		
XC22	Orange	4/$1	XC526	Orange	4/$1	XC556	Orange	4/$1	¼" x ¼" x 1/16"		
SSL-22	RT	4/$1	XC526	Clear	4/$1	XC556	Clear	7/$1	Flat 5/$1.00		

SPECIAL * — XC556 Red 100/$8.00 1000/$60.00 — SPECIAL *

DISPLAY LEDS

DL707   DL33B

TYPE	POLARITY	HT		TYPE	POLARITY	HT	
MAN 1	Common Anode	.270	2.95	MAN 3640	Common Cathode-orange	.300	1.75
MAN 2	5 x 7 Dot Matrix	.300	4.95	MAN 4710	Common Anode-Red	.400	1.95
MAN 3	Common Cathode	.125	3/1.00	DL701	Common Anode-red ±	.300	.99
MAN 4	Common Cathode	.187	1.95	DL704	Common Cathode	.300	.99
MAN 7	Common Anode	.300	1.25	DL707	Common Anode	.300	.99
MAN 7G	Common Anode-green	.300	1.95	MAN 4740	Common Anode-Red	.400	.99
MAN 7Y	Common Anode-yellow	.300	1.95	DL741	Common Anode	.600	1.50
MAN 52	Common Anode-green	.300	.99	DL 747	Common Anode	.600	2.25
MAN 64	Common Anode-red	.400	.99	DL 750	Common Cathode	.600	2.49
MAN 74	Common Cathode	.300	1.50	DL 338	Common Cathode	.110	.50
MAN 82	Common Anode-yellow	.300	.99	FND70	Common Cathode	.250	.75
MAN 84	Common Cathode-yellow	.300	.99	FND503	Common Cathode	.500	1.00
MAN 3620	Common Anode -orange	.300	1.75	FND507	Common Anode	.500	1.00

FACTORY REJECTS ATARI GAME BOARDS

BOARD A — 8½" x 16" **$6.95 ea.**
Over 60 each reusable IC's
Misc. Transistors, Resistors, Diodes, Caps, Crystals, Switch, etc.
ONLY 500 EA. AVAILABLE

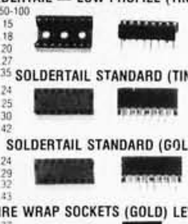
DELUXE BOARD B — 11½" x 18" **$9.95 ea.**
Over 100 each reusable IC's
Misc. Tranistors, Resistors, Diodes, Caps, Crystals, Switches, LEDS, etc.
ONLY 500 EA. AVAILABLE

HP 5082-7300 Multi-Digit Series

- ⅛" Ht. • Common Cathode • Dip Package
- 3 to 5 volts @5 mils per segment
- 7 segment Monolithic • Red Display

2 Digit	$ .79
3 Digit	.89
4 Digit	.99
5 Digit	1.19

IC SOLDERTAIL — LOW PROFILE (TIN) SOCKETS



	1-24	25-49	50-100		1-24	25-49	50-100
8 pin	$.17	.16	.15	24 pin	$.38	.37	.36
14 pin	.20	.19	.18	28 pin	.45	.44	.43
16 pin	.22	.21	.20	36 pin	.60	.59	.58
18 pin	.29	.28	.27	40 pin	.63	.62	.61
22 pin	.37	.36	.35				

SOLDERTAIL STANDARD (TIN)

14 pin	$.27	.25	.24	28 pin	$ .99	.90	.81
16 pin	.30	.27	.25	36 pin	1.39	1.26	1.15
18 pin	.35	.32	.30	40 pin	1.59	1.45	1.30
24 pin	.49	.45	.42				

SOLDERTAIL STANDARD (GOLD)

8 pin	$.30	.27	.24	24 pin	$ .70	.63	.57
14 pin	.35	.32	.29	28 pin	1.10	1.00	.90
16 pin	.38	.35	.32	36 pin	1.75	1.40	1.26
18 pin	.52	.47	.43	40 pin	1.75	1.59	1.45

WIRE WRAP SOCKETS (GOLD) LEVEL #3

10 pin	$.45	.41	.37	24 pin	$1.05	.95	.85
14 pin	.39	.38	.37	28 pin	1.40	1.25	1.10
16 pin	.43	.42	.41	36 pin	1.59	1.45	1.30
18 pin	.75	.68	.62	40 pin	1.75	1.55	1.40

Plastic Push Button Switch

- 18 AWG Solid Wire - 5" Long
- .50 (wide) X .60 (high) ⅜-27 Thread
- 8 AMP @ 14 Volt - 1 AMP @ 110 Volt

		1-9	10-Up
J-188-1	Push On-Push Off	.59	.49
J-188-2	Normally Open	.59	.49
J-188-3	Normally Closed	.59	.49

MINATURE TOGGLE SWITCH

JMT-221	DPDT	on/off/on	$1.95
JMT-223	DPDT	on/none/on	$1.75
JMT-121	SPDT	on/off/on	$1.50
JMT-123	SPDT	on/none/on	$1.25

CLIPLITE 8/$1.49

LED MOUNTING SYSTEM
use with XC556 LEDS
Specify Colors — Red - Green - Amber - Yellow

DIP SWITCHES

SPST Slide Action

#206-4 (8 pin dip) 4 switch unit $1.75 ea.
#206-7 (14 pin dip) 7 switch unit $1.95 ea.
#206-8 (16 pin dip) 8 switch unit $2.25 ea.

50 PCS. RESISTOR ASSORTMENTS $1.75 PER ASST.

ASST. 1	5 ea.	10 OHM 27 OHM	12 OHM 33 OHM	15 OHM 39 OHM	18 OHM 47 OHM	22 OHM 56 OHM	1/4 WATT 5% 50 PCS.
ASST. 2	5 ea.	68 OHM 180 OHM	82 OHM 220 OHM	100 OHM 270 OHM	120 OHM 330 OHM	150 OHM 390 OHM	1/4 WATT 5% 50 PCS.
ASST. 3	5 ea.	470 OHM 1.2K	560 OHM 1.5K	680 OHM 1.8K	820 OHM 2.2K	1K 2.7K	1/4 WATT 5% 50 PCS.
ASST. 4	5 ea.	3.3K 8.2K	3.9K 10K	4.7K 12K	5.6K 15K	6.8K 18K	1/4 WATT 5% 50 PCS.
ASST. 5	5 ea.	22K 56K	27K 68K	33K 82K	39K 100K	47K 120K	1/4 WATT 5% 50 PCS.
ASST. 6	5 ea.	150K 390K	180K 470K	220K 560K	270K 680K	330K 820K	1/4 WATT 5% 50 PCS.
ASST. 7	5 ea.	1M 2.7M	1.2M 3.3M	1.5M 3.9M	1.8M 4.7M	2.2M 5.6M	1/4 WATT 5% 50PCS.

ASST. 8R Includes Resistor Assortments 1-7 (350 PCS.) **$10.95 ea.**

$5.00 Minimum Order — U.S. Funds Only
California Residents — Add 6% Sales Tax
Spec Sheets - 25¢ — Send 35¢ Stamp for 1977A Catalog
Dealer Discount Available — Request Pricing

NOW OPEN SATURDAYS

James ELECTRONICS

1021-A HOWARD AVE., SAN CARLOS, CA. 94070
PHONE ORDERS WELCOME — (415) 592-8097
All Advertised Prices Good Thru June

WIRE WRAP CENTER

HOBBY-WRAP TOOL-BW-630

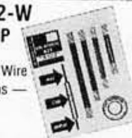
- Battery Operated (Size C)
- Weighs ONLY 11 Ounces
- Wraps 30 AWG Wire onto Standard DIP Sockets (.025 inch)

Complete with built-in bit and sleeve

$34.95
(batteries not included)

WIRE-WRAP KIT — WK-2-W

WRAP • STRIP • UNWRAP



- Tool for 30 AWG Wire
- Roll of 50 Ft. White or Blue 30 AWG Wire
- 50 pcs. each 1", 2", 3" & 4" lengths — pre-stripped wire.

$11.95

WIRE WRAP TOOL WSU-30

WRAP • STRIP • UNWRAP - $5.95

WIRE WRAP WIRE — 30 AWG

25 ft. min. $1.25 50 ft. $1.95 100 ft. $2.95 1000 ft. $15.00
SPECIFY COLOR — White - Yellow - Red - Green - Blue - Black

CUTTER CRIMPER TOOL (CS-8)

- Plier Nose (serrated-jaw)
- Scissors Action Cutting
- 6 Bolt Cutters (4-40, 5-40, 6-32, 8-32, 10-32, 10-24)
- Crimp Stations (7mm Auto — 22-20 to 12-X0 elect.)
- "Up-Front" Wire Cutting
- Scissors Action Stripping (No. 22-20 to No. 10)
- Crimp Stations — insulated (2w-20 to 12-10 elect.)

Actual Size - 8¼" length **$8.50**

Permacel Electrical Tape

- ¾" (wide) X 66 ft. (long) • All weather • Not import

$1.25 per roll — $9.95 per 10 roll package

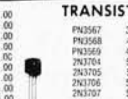
ZENERS — DIODES — RECTIFIERS

TYPE	VOLTS	W	PRICE	TYPE	VOLTS	W	PRICE
1N746	3.3	400mm	4/1.00	1N4005	600 PIV	1 AMP	10/1.00
1N751A	5.1	400m	4/1.00	1N4006	800 PIV	1 AMP	10/1.00
1N752	5.6	400m	4/1.00	1N4007	1000 PIV	1 AMP	10/1.00
1N753	6.2	400m	4/1.00	1N3600	50	200m	6/1.00
1N754	6.8	400m	4/1.00	1N4148	75	10m	15/1.00
1N959	8.2	400m	8/1.00	1N4154	35	10m	12/1.00
1N965B	15	400m	4/1.00	1N4305	75	25m	20/1.00
1N5232	5.6	500m	.28	1N4734	5.6	1w	.28
1N5234	6.2	500m	.28	1N4735	6.2	1w	.28
1N5235	6.8	500m	.28	1N4736	6.8	1w	.28
1N5236	7.5	500m	.28	1N4738	8.2	1w	.28
1N456	25	40m	6/1.00	1N4742	12	1w	.28
1N458	150	7m	6/1.00	1N4744	15	1w	.28
1N485A	180	10m	6/1.00	1N1183	50 PIV	35 AMP	1.60
1N4001	50 PIV	1 AMP	12/1.00	1N1184	100 PIV	35 AMP	1.70
1N4002	100 PIV	1 AMP	12/1.00	1N1185	150 PIV	35 AMP	1.50
1N4003	200 PIV	1 AMP	12/1.00	1N1186	200 PIV	35 AMP	1.80
1N4004	400 PIV	1 AMP	12/1.00	1N1188	400 PIV	35 AMP	3.00

SCR AND FW BRIDGE RECTIFIERS

C36D	15A @ 400V	SCR	$1.95
C38M	35A @ 200V	SCR	1.95
2N2328	1.6A @ 200V	SCR	.50
MDA 980-1	12A @ 50V	FW BRIDGE REC.	1.95
MDA 980-3	12A @ 200V	FW BRIDGE REC.	1.95

TRANSISTORS



MPS A05	5/$1.00			PN4249	4/$1.00
MPS A06	5/$1.00			PN4250	4/$1.00
2N2219A	3/$1.00	PN3567	3/$1.00	2N4400	4/$1.00
2N2221	4/$1.00	PN3568	4/$1.00	2N4401	4/$1.00
2N2222A	5/$1.00	PN3569	4/$1.00	2N4402	4/$1.00
2N2369	5/$1.00	2N3704	5/$1.00	2N4403	4/$1.00
2N2369A	4/$1.00	2N3705	5/$1.00	2N4409	5/$1.00
2N2484	4/$1.00	2N3706	5/$1.00	2N5086	4/$1.00
2N2906A	4/$1.00	2N3707	5/$1.00	2N5087	4/$1.00
2N2907A	5/$1.00	2N3711	5/$1.00	2N5088	4/$1.00
2N2925	5/$1.00	2N3724	$ .65	2N5089	4/$1.00
2N3053	2/$1.00	2N3725	$1.00	2N5129	5/$1.00
2N3055	$ .89	2N3903	5/$1.00	2N5138	5/$1.00
MJE3055	$1.00	2N3904	4/$1.00	2N5139	5/$1.00
MJE2955	$1.25	2N3905	4/$1.00	2N5209	5/$1.00
2N3392	5/$1.00	2N3906	4/$1.00	2N5951	5/$1.00
2N3398	5/$1.00	2N4013	3/$1.00	C106B1SCR	2/$1.00
		2N4014	3/$1.00	2N5432	$2.00
		2N4123	10/$1.00		

CAPACITOR CORNER

50 VOLT CERAMIC DISC CAPACITORS

	1-9	10-49	50-100		1-9	10-49	50-100
10 pf	.05	.04	.03	.001µF	.05	.04	.035
22 pf	.05	.04	.03	.0047µF	.05	.04	.035
47 pf	.05	.04	.03	.01µF	.05	.04	.035
100 pf	.05	.04	.03	.022µF	.06	.05	.04
220 pf	.05	.04	.03	.047µF	.06	.05	.04
470 pf	.05	.04	.035	.1µF	.12	.09	.075
100 VOLT MYLAR FILM CAPACITORS							
.001mf	.12	.10	.07	.022mf	.13	.11	.08
.0022	.12	.10	.07	.047mf	.21	.17	.13
.0047mf	.12	.10	.07	.1mf	.27	.23	.17
.01mf	.12	.10	.07	.22mf	.33	.27	.22
+20% DIPPED TANTALUMS (SOLID) CAPACITORS							
.1/35V	.28	.23	.17	1.5/35V	.30	.26	.21
.15/35V	.28	.23	.17	2.2/25V	.31	.27	.22
.22/35V	.28	.23	.17	3.3/25V	.31	.27	.22
.33/35V	.28	.23	.17	4.7/25V	.32	.28	.23
.47/35V	.28	.23	.17	6.8/25V	.36	.31	.25
.68/35V	.28	.23	.17	10/25V	.40	.35	.29
1.0/35V	.28	.23	.17	15/25V	.63	.50	.40
MINIATURE ALUMINUM ELECTROLYTIC CAPACITORS							
Axial Lead				Radial Lead			
.47/50V	.15	.13	.10	.47/25V	.15	.13	.10
1.0/50V	.16	.14	.11	.47/50V	.16	.14	.11
3.3/50V	.15	.13	.10	1.0/16V	.15	.13	.10
4.7/25V	.16	.14	.12	1.0/25V	.16	.14	.11
10/25V	.15	.13	.10	1.0/50V	.16	.14	.11
10/50V	.16	.14	.12	4.7/16V	.15	.13	.10
22/25V	.17	.15	.12	4.7/25V	.15	.13	.10
22/50V	.24	.20	.18	4.7/50V	.16	.14	.11
47/25V	.19	.17	.15	10/16V	.14	.12	.09
47/50V	.25	.21	.19	10/25V	.15	.13	.10
100/25V	.24	.20	.18	10/50V	.16	.14	.12
100/50V	.35	.30	.28	47/50V	.24	.21	.19
220/25V	.32	.28	.25	100/16V	.19	.15	.14
220/50V	.45	.41	.38	100/25V	.24	.20	.18
470/25V	.33	.29	.27	100/50V	.35	.30	.28
1000/16V	.55	.50	.45	220/16V	.23	.17	.16
2200/16V	.70	.62	.55	470/25V	.31	.28	.26

Circle 15 on inquiry card.

computer display terminal

This display terminal has an integral controller, B/W cathode ray tube and keyboard. The system has a serial I/O interface for communication and an I/O interface for a printer

DISPLAY (P/N 4802-1095-501) FEATURES:
- 17" B/W CRT
- 41 lines of data
- 52 characters per line
- Characters are generated by a diode matrix "graphic" technique
- 21 special push-buttons wired for a program call up
- Brightness Control
- Self-contained power supply

KEYBOARD (P/N 4802-1115-501) FEATURES:
- Reed switch technology
- 54 data keys
- 28 special keys detachable with cable

LOGIC UNIT (P/N 4802-1157-502) FEATURES:
- 1024 by 6 bit core memory
- Printer I/O interface
- Communication I/O interface

POWER: 115V, 50/60 Hz, 500 Watts

WEIGHT: 210 lbs. (including logic unit, keyboard, display and cables.)

FOB LYNN MASS (you pay shipping)
Check with order please.

External logic & power pack not shown.

$180.00

"AS IS"

4 way cursor control, graphics display.

The story: These are unused terminals made for airport ticketing & seat assignment. After several years of storage they require tinkering to make operable. We have some hints printed such as cleaning PC fingers. One of our customers has this tied into his KIM-1, another has his running with his IMSAI. We have data on this. Should be useable on most common computers. A hell of a deal and all for a paltry $180.00. Don't be left out as many were on our past VIATRON deal. Sold "as is" all sales final.

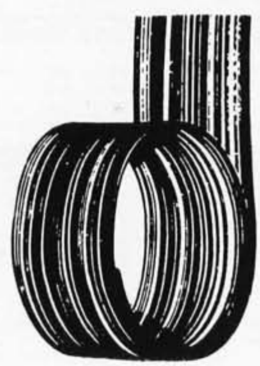
WITH COMPLETE DOCUMENTATION

SPECTRA FLAT TWIST

50 conductor, 28 gauge, 7 strands/conductor made by Spectra. Two conductors are paired & twisted and the flat ribbon made up of 25 pairs to give total of 50 conductor. May be peeled off in pairs if desired. Made twisted to cut down on "cross talk." Ideal for sandwiching PC boards allowing flexibility and working on both sides of the boards. Cost originally $13.00/ft

SP-324-A $1.00/ft. 10 ft/$9.00

SP-234-A $1.00 ft 50 cond. 10 ft/$9.00
SP-234-B .90 ft 32 cond. 10 ft/$8.00



WIRE WRAP WIRE

TEFZEL blue #30 Reg. price $13.28/100 ft. Our price 100 ft $2.00; 500 ft $7.50.

MULTI COLORED SPECTRA WIRE

Footage			10'	50'	100'
8	Cond.	#24	$2.50	9.00	15.00
12	"	22	3.00	11.00	18.00
14	"	22	3.50	13.00	21.00
29	"	22	7.50	28.00	45.00

Great savings as these are about 1/4 book prices. All fresh & new.

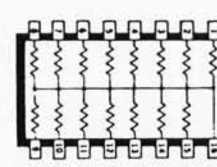
In tall TO-5 can DPDT, 24 volts. Brand new. cost $16.00 each

TINY SWITCH TELEDYNE

SP-134 $3.00 each 2/$5.00

Precision 16 pin DIP network as shown. Each resistor 1K. For pull-up/pull-down interface networks. Value over $1.00 each; New, CTS or Beckman

SP-320 pack of 6 $1.00



Meshna

Please add shipping cost on above. Minimum order $10
FREE CATALOG SP-9 NOW READY
P.O. Box 62, E. Lynn, Massachusetts 01904

Circle 18 on inquiry card.

KITS BY cybercom

4Kx8 Static Memories
MB-1 MK-8 board, 1 usec 2102 or eq. PC Board $22 (Not for S-100 Buss)

MB-4 Improved MB-2 designed for 8K "piggy-back" without cutting traces. PC Board $30
Kit 4K .5 usec....$129.95 Kit 8K .5 usec....$199

MB-3 1702A's Eroms, Altair 8800 & Imsai 8080 compatible switched address & wait cycles. 2K may be expanded to 4K. Kit less Proms....$65 2K Kit....$105 4K Kit....$145

MB-6 PC Board $35

MB-6A 8Kx8 Switched address and wait assignments. Memory protection is switchable for 256, 512, 1K, 2K, 4K and 8K. 91L02A .5 usec rams. Altair 8800 & Imsai compatible. With battery power option.
Kit....$250 Assembled & tested $290

I/O Boards
I/O-2 I/O for 8800, 2 ports, committed pads for 3 more, other pads for EROMS UART, etc.
Kit....$55 PC Board only $25

64 x 16 VIDEO BOARD Altair plug compatible display 32 x 16 or 64 x 16 switch selectable. Composite and parallel video ports, upper and lower case with software.
Kit $179.95
PC Board $35

SP-1 Synthesizer Board Computer controlled wave forms 9 octaves 1vrms ½% distortion includes high level music langauge. Kit $250

Altair Compatible mother board. Room for 15 connectors 11" x 11½" (w/o connectors) $45
With 15 connectors $105

Altair Extender Board (w/o connectors) $9.00
With w/w connector $13.50

100 Pin spec WW or Soldertail both fit Imsai or SSM Mother Board $5.00 each 10/$44.

1702A EROM	$ 8.00
programming send hex list	5.00
AY5-1013 UART	6.95
2513 Prime spec. upper or lower case	11.00
8080A prime CPU	25.00
8212 prime latch buffer	4.00
8224 prime clock gen	5.00
8228 prime sys controller	8.90

82S06	2.00	82S126	3.50	74C200	5.50
82S07	2.00	82S129	3.50	8573	4.50
82S11	2.00	82S130	3.95	8574	5.50
82S12	2.00	82S131	3.95	8575	4.50
82S17	2.00	74S206	2.10	8576	4.50
82S23	2.50	74S412	4.00	8577	3.50
82S123	3.00			8578	4.00

Kits by WALCON
(AVAILABLE MID-MAY)

MM-1 8Kx8 Ram Board First fully buffered 8K ram board. Addresses and data are fully buffered. Low power rams used. Protect is reset with a single switch. Documentation is available for $5 which is refunded at time of purchase of board or kit.
MM-1 Kit $245
Assembled 295
PC Board only 30

★ **2102AL-4 FULL SPEED MEMORY (500ns)** ★
★ **Nearly ⅓ less power consumption of even a 21L02A** ★ **PRIME FROM NEC**
★ **$2.00 EACH** **32 FOR $1.80 EACH** ★
64 - $1.70 **128 FOR $1.60 EACH**

7400	.16	7470	.45	74161	1.00
7401	.16	7472	.40	74162	1.50
7402	.21	7473	.35	74163	1.00
7403	.16	7474	.35	74164	1.10
7404	.18	7475	.50	74165	1.10
7405	.24	7476	.30	74166	1.25
7406	.20	7480	.50	74170	2.10
7407	.29	7483	.70	74173	1.50
7408	.25	7485	.90	74174	1.95
7409	.25	7486	.40	74175	.95
7410	.18	7489	2.00	74176	.90
7411	.30	7490	.45	74177	.90
7413	.45	7491	.75	74179	.90
7414	.70	7492	.50	74180	.90
7416	.35	7493	.50	74181	2.50
7417	.35	7494	.80	74182	.95
7420	.20	7495	.75	74184	1.95
7423	.37	7496	.90	74185	2.20
7425	.30	74100	1.00	74190	1.15
7426	.30	74107	.40	74191	1.25
7427	.35	74109	.90	74192	.90
7430	.25	74121	.40	74193	.90
7432	.30	74122	.50	74194	1.25
7437	.27	74123	.70	74195	.75
7438	.27	74125	.60	74196	1.25
7440	.15	74126	.60	74197	.90
7441	.85	74132	1.00	74198	1.75
7442	.60	74141	1.15	74199	1.75
7443	.75	74145	1.15	74200	4.95
7444	.75	74147	2.35	74251	1.75
7445	.75	74148	2.00	72284	4.95
7446	.80	74150	1.00	74285	4.95
7447	.70	74151	.80	74365	.90
7448	.80	74153	.90	74367	.75
7450	.25	74154	1.00	74368	.90
7451	.25	74155	1.00	MH0025	2.50
7453	.25	74156	1.00	MH0026	2.95
7454	.20	74157	1.00	95H90	9.95
7460	.20	74160	1.25		

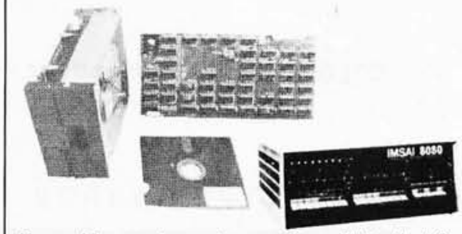
MM5309	8.00	2501B	$1.25	1101	.40	74LS00	.40	74L00	.25	74L78	.90
MM5312	4.00	2503V	2.00	1103	1.25	74LS01	.50	74L01	.25	74L85	1.40
MM5313	4.00	2504V	2.00	2101	4.50	74LS02	.40	74L02	.25	74L86	.75
MM5320	4.95	2505KN	2.00	2112	4.50	74LS03	.40	74L03	.25	74L89	3.50
MM5554	1.90	2507V	1.25	2602	1.60	74LS04	.45	74L04	.30	74L90	1.50
MM5556	2.50	2509A	2.00	4002-1	7.50	74LS05	.45	74L05	.40	74L91	1.50
MM5055	1.90	2510A	2.00	4002-2	7.50	74LS10	.40	74L06	.30	74L93	1.70
DM8836	.60	2511A	2.80	MM5262	1.00	74LS11	.50	74L08	.40	74L95	1.70
DM8837	1.50	2517V	1.25	7489	2.00	74LS12	.55	74L09	.40	74L98	2.80
80C95	1.10	2518B	1.50	74200	4.95	74LS20	.40	74L10	.30	74L123	1.50
80C97	1.00	2519B	2.80	74S89	3.50	74LS22	.45	74L20	.35	74L154	2.00
80L97	1.50	2521V	1.50	74C89	3.50	74LS27	.45	74L26	.40	74L164	2.50
81L22	1.50	2522V	2.00	74L89	3.50	74LS30	.40	74L30	.40	74L165	2.50
81L23	1.90	2525V	2.80	8T80	2.50	74LS42	1.50	74L32	.45	74L192	1.25
81L51	2.50	2527V	2.80	8T97	2.00	74LS55	.40	74L42	1.50	74L193	1.20
85L52	2.50	2528V	2.80	INTEL		74LS73	.65	74L51	.35	MC4044	2.25
85L63	1.25	2529V	2.80	8216	4.95	74LS74	.65	74L54	.45	N8263	3.50
86L70	1.50	2532B	2.80	8214	8.30	74LS76	.65	74L55	.35	N8826	2.50
86L75	1.90	2533V	2.80	8251	14.50	74LS151	1.55	75L71	.30	DM8131	2.50
86L99	3.50	91L02APC	2.55	8255	14.50	74LS174	2.20	74L73	.55	8T16	2.00
86L12	.80	32 ea.	2.40	1488	1.50	74LS175	1.95	74L74	.55	8T20	2.00
8T13	2.50	64 ea.	2.25	1489	1.50	74LS192	2.85	74L75	1.20	8T10	2.00

MIKOS
419 Portofino Drive
San Carlos, California 94070
Please send for xistor, IC and kit list

For large orders please send money order or cashier's check to avoid delays in waiting fr checks to clear.

Check or money order only. California residents add 6% tax. All orders postpaid in U.S. All devices tested prior to sale. Money back 30 day guarantee. Sorry we cannot accept returned ICs that have been soldered to. $10 minimum order. Prices subject to change without notice.

The Proko Paper Tape Reader
A fully TTL buffered optical tape reader for reading 8-level paper tape. It's compatible with nearly all parallel input ports and has both positive and negative strobe outputs. The power (+5V @ 20ma) is derived from your I/O board...just supply a light source (study lamp), grab the tape and pull!
PTR-II Kit $54.00 **Assembled $68.00**

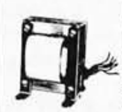


Now, add mass storage to your micro with the North Star floppy disk system. Comes with cables, S-100 controller with bootstrap rom, and powerful extended basic. (Basic supports sequential and random access disk files.)
North Star Disk Kit $699
IMSAI 8080 Kit with 22 slots $650

SPECIAL DEAL: IMSAI WITH DISK $1325

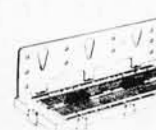
The S.D. Sales Co. Z-80 CPU Kit
For S-100 Buss
$149.00

SPECIAL DEAL No. 2
IMSAI 8080 with 22-SLOT MB UPGRADED WITH Z-80 CPU $750.00



POWER TRANSFORMER FOR ALTAIR® TYPE BUSS
Designed to deliver +18VDC @ 2A and 5VDC @ 22A with proper bridges and filtering.
29.50 each

NEW FROM IMSAI
32K RAM CARDS for S-100 Buss
Full Speed — Low, Low Power
$725.00



Bimboards for Prokotyping
$13.50 each
4 for $44.00

D-CONNECTORS
25-Pin

Male $3.00
Female $4.00
Pair $6.50

proko tronics
439 marsh street
san luis obispo, california 93401
805/544/5441

Delivery: Stock to 45 days from receipt of order. We pay all U.P.S. shipping on U.S. orders. Check or money order accepted. No P.O.s except by above terms. California residents add 6% tax. Prices subject to change without notice.

New Component Values

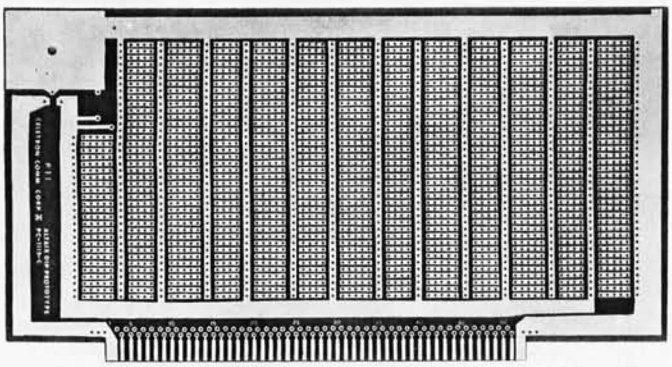
PRIME, NEW 2708 1K x 8 EPROM – $ 47.95

PRIME, NEW 2716 2K x 8 EPROM – $ 77.95

ZILOG Z-80's $ 47.95

91L02C's, 300ns low-power – $ 1.89

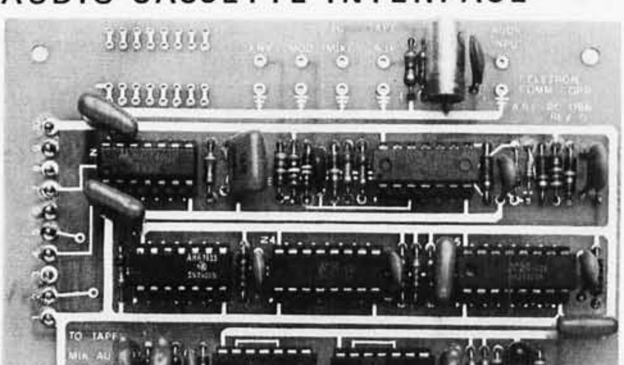
PROTOTYPE BOARDS



Prototype boards for the S-100 bus are available from many others —but only MINI MICRO MART supplies four different types. Two are wire-wrap versions and two are general-purpose DIP, for either ww or point-to-point wiring. All boards come with a 5V regulator and a heat sink. The two "bus" versions are unique and have circuitry etched on for buffering and address decoding, and include the decoders and necessary tri-state buffers. (Illustrated above is the general-purpose DIP version, MODEL 01-2115.)

01-2115	General-purpose DIP Prototype Board	$ 18.95
01-2116	Wire-wrap Prototype Board	$ 19.95
01-2136	Gen.-purpose DIP Bus Interface Bd...	$ 29.95
01-2112	Wire-wrap Bus Interface Board	$ 30.95

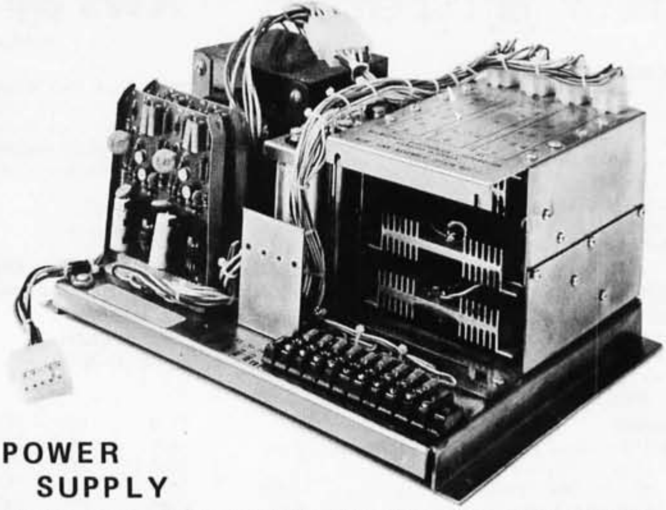
AUDIO CASSETTE INTERFACE



This simple board can be used with any minicomputer or TV terminal that uses a UART with a 16X baud-rate clock. Designed for the St. Louis BYTE standard, it also provides for a tone/no-tone and a HITS interface. Available in kit form, order as –

51-2166 Audio Cassette Interface .... $ 24.95

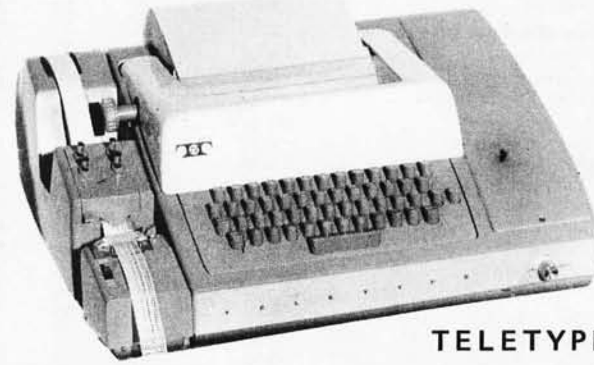
Surplus Bargains



POWER SUPPLY

Surplus power supply (made by Milwaukee Electronics) removed from used Mohawk Data equipment, excellent condition..... 'Sorry we can't give full details at press time, but we're sure that it will supply all the +5 you'll ever need as well as other voltages. 'Has all the good things – such as over-voltage protection. We hope to have schematics to ship with these units. Limited supply. A REAL VALUE! Order as –

42-5005 Power Supply .... $ 24.95



TELETYPES

MINI MICRO MART has one of the largest selections of used, reconditioned, and rebuilt Teletypes in the U.S. –

RO-33's (printer only)	$395 to $595
KSR-33's (keyboard & printer)........	$495 to $695
ASR-33's (prntr.,keybd.,reader & punch) .	$695 to $895

Model 35 RO's, KSR's and ASR's also available.

SURPLUS PERIPHERALS

We have a tremendous inventory of used and like-new computer peripherals which we have rescued from large computer systems. Among them are: Delta video terminals, Hazeltine 1000 video terminals, Univac 30 CPS serial printers, Singer 30 CPS serial printers, 9-track digital reel-to-reel recorders, used commercial minicomputers and intelligent terminals.

Also available are a wide variety of transformers, power-supply kits, and surplus power supplies.

Send stamped, self-addressed envelope for details on any advertised items or for a copy of our catalog.

Add $2 for handling, shipping & insurance for each item ordered.

MiniMicroMart

1618 James Street, Syracuse, N.Y. 13203, Phone: (315) 422-4467

Circle 279 on inquiry card.

Circle 32 on inquiry card.

Reader Service

To get further information on the products advertised in BYTE, fill out the reader service card with your name and address. Then circle the appropriate numbers for the advertisers you select from this list. Add a 9 cent stamp to the card, then drop it in the mail. Not only do you gain information, but our advertisers are encouraged to use the marketplace provided by BYTE. This helps us bring you a bigger BYTE.

Reader Service Number	Page Number	Reader Service Number	Page Number	Reader Service Number	Page Number
75	Advanced Microcomputer 164	157	Electronic Warehouse 168	260	Objective Design Inc 146
168	Aldelco 165	247	Electravalue 159	40	Ohio Scientific Instruments 49
255	Alpha Digital Systems 151	102	Eltron 169	64	Oliver Audio Engineering 161
272	Apple Computer 14, 15, 17	257	Fein-Marquart 159	198	Omni 147
275	Artison Computer 128	248	Franklin Electric 67	*	PC 77 Atlantic City 75
276	Beta Business Systems 151	9	Godbout Electronics 170	273	Page Digital Electronics 163
4	Bits Inc 131, 132, 162, 163	*	HAL Communications 129	265	Paia Electronics Inc 23
*	Byte Index 127	262	Hashizume Burt 161	63	Parasitic 109, 110
227	Byte Shop East 151	151	Heath Co 68, 69, 70, 71	*	PerCom Data 12, 72
271	Byte Shop of Miami 127	12	IMSAI 5	194	Peripheral Vision 29
200	California Industrial 166	117	Intel 10, 11	281	Perri-White 110
*	Camp Retupmoc 151	204	International Data Systems 160	*	Personal Computing Expo 149
127	Comptek 147	268	IOR 161	*	PolyMorphic Systems CIII
38	Compucolor 60,61	278	Ithaca Audio 145	213	Prime Radix 153
140	Computalker 151	215	Jade Company 171	24	Processor Technology 6, 7, 8
161	Computer Corner 159	15	James 172, 173	219	RHS Marketing 98, 99
253	Computer Electronics 127	214	John Wiley & Sons 105	201	Riverside Electronic 158
143	Computer Enterprises 148	209	Logic Design 127	234	Rotundra Cybernetics 163
176	Computer Mart NH 159	*	MACC 130	27	SD Sales 177
83	Computer Mart NY 162	18	Meshna 174	282	Sams Howard W 51
156	Computer Place 159	261	Micro Computers 161	26	Scelbi 25
141	Computer Room 115	77	Micro-Term 107	73	Scientific Research 79
208	Computer Shack 111	242	Micromation 26	236	SEALS 73
179	Computer Shop (Canada) 127	196	Micronics 161	169	Smoke Signal Broadcasting 120
138	Computer Warehouse Store 167	119	Midwestern Scientific Inst 121	59	Solid State Sales 178
6	Continental Specialties 126	57	Mikos 175	29	Southwest Technical Products CII
202	CRC Engineering 159	279	Mini Micro Mart 176	164	Sunny Computer Stores 163
87	Creative Computing 143	112	MiniTerm 122, 123, 124, 125	96	Synchro-Sound Enterprises 86, 87
41	Cromemco 1, 2	*	MITS CIV	205	Szerlip Enterprises 108
178	Cybercom 13	250	Mountain Hardware 151	121	Tarbell Electronics 146
223	Data Search 108	71	mpi 145	82	Technical Design Labs 53
78	Digital Group 28	171	Mullen 144	136	Technical Systems Consultants 107
263	Digital Concepts 144	264	MVT Microcomputer Systems 161	32	Tri-Tek Inc 179
56	E&L Instruments 113	*	NCC 63, 64, 65, 66	270	Urban Instruments 112
79	Economy Company 139	155	North Star Computers 24	137	Vector Graphic Inc 74
269	Edityper Systems Corp 114	280	Northern Valley Systems 148	220	Ximedia 27
47	Electronic Control Tech 127	277	Noval 81, 82, 83, 84	222	Xybek 114

***Reader service inquiries not solicited. Correspond directly with company.**

BOMB— BYTE's Ongoing Monitor Box

On BOMB Card, Article No.	ARTICLE	PAGE
1	**Kraul: Designing Multichannel Analog Interfaces**	18
2	**Hollis: Newt: A Mobile, Cognitive Robot**	30
3	**Fylstra: Interfacing the IBM Selectric Keyboard Printer**	46
4	**Carr: Interfacing to an Analog World: Part 2**	54
5	**Simpson: Come Fly with KIM**	76
6	**Welles: Software for the Economy Floppy Disk**	88
7	**Wimble: Artificial Intelligence: Part 2, Implementation**	100
8	**Quek: Introduction to Microprogramming**	116
9	**Guzzon: A 6800 Selectric IO Printer Program**	140
10	**McNatt: A Guide to Baudot Machines: Part 3**	154

The Bomb of March

The BOMB of March has fallen. In the tabulation of the stacks of cards returned with article evaluations, Jack Breimeir and Ira Rampil came out first, and receive the $100 prize for their article "The Digital Cassette Subsystem, Part 2." Second prize winner in the tally came up as a tie between Steve Ciarcia's "Try This Computer on for Size" and Thomas R Buschbach's "An Inexpensive Joystick Interface." Steve and Thomas will each receive a $50 bonus check.■

The POLY 88 Microcomputer System

PolyMorphic Systems now offers the complete, assembled, personal computer system—the POLY 88 System 16. A full 16K system with high speed video display, alphanumeric keyboard, and cassette program storage. A BASIC software package providing the most advanced features available in the personal computing market. Features like PLOT and TIME, which utilize our video graphics and real-time clock. Others like VERIFY, so that you know your tape is good before you load another. Or input type-ahead so you can tell your program to run while the tape is still loading (it stores up to 64 characters of commands or question responses to be executed). All these plus a complete package of scientific functions, formatting options, and string capabilities. With the POLY 88 System 16 you can amaze your timesharing friends the very first night!

Polymorphic Systems 11K BASIC — Size: 11K bytes.

Scientific Functions: Sine, cosine, log, exponential, square root, random number, x to the y power.

Formatted Output • Multi-line Function Definition • String Manipulation and String Functions • Real-Time Clock • Point-Plotting on Video Display • Array dimensions limited by memory • Cassette Save and Load of Named Programs • Multiple Statements per Line • Renumber • Memory Load and Store • 8080 Input and Output • If Then Else • Input type-ahead.

Commands: RUN, LIST, SCR, CLEAR, REN, CONTINUE

Statements: LET, IF, THEN, ELSE, FOR, NEXT, GOTO, ON, EXIT, STOP, END, REM, READ, DATA, RESTORE, INPUT, GOSUB, RETURN, PRINT, POKE, OUT.

Built in Functions: FREE, ABS, SGN, INT, LEN, CHR$, VAL, STR$, ASC, SIN, COS, RND, LOG, TIME, WAIT, EXP, SQRT, CALL, PEEK, INP, PLOT.

Systems Available. The POLY 88 is available in either kit or assembled form. It is suggested that kits be attempted only be persons familiar with digital circuitry.

System 2: is a kit consisting of the POLY 88 chassis, CPU, video circuit card, and cassette interface. Requires keyboard, TV monitor, and cassette recorder for operation. $735

System 16: consists of an assembled and tested System 2 with 16K of memory, keyboard, TV monitor, cassette recorder, 11K BASIC and Assembler on cassette tapes. $2250.

System 0: The circuit cards an S-100 mainframe owner needs to be compatible with the POLY 88 software library. System 0 consists of the central processor card with monitor ROM, the video circuit card, and cassette interface, all in kit form. $525.

Prices and specifications are subject to change without notice.
California residents add 6% sales tax.

460 Ward Drive
Santa Barbara, Ca. 93111
(805) 967-2351

PolyMorphic Systems

Now you can buy an Altair™ 8800b or an Altair 680b computer right off the shelf. Altair plug-in boards, peripherals, software and manuals are also available. Check the list below for the MITS dealer in your area.

off the shelf.

ALTAIR COMPUTER CENTER
4941 East 29th St
TUCSON, AZ 85711
(602)-748-7363

COMPUTER KITS
1044 University Ave
BERKELEY, CA 94710
(415)-845-5300

THE COMPUTER STORE
820 Broaoway
SANTA MONICA, CA 90401
(213)-451-0713

GATEWAY ELECTRONICS, INC
OF COLORADO
2839 W 44th Ave
DENVER, CO 80211
(303)-458-5444

THE COMPUTER STORE, INC
(Hartford area)
63 South Main Street
WINDSOR LOCKS, CT 06096
(203)-627-0188

MARSH DATA SYSTEMS
5405 B Southern Comfort Blvd
TAMPA, FL 33614
(813)-886-9890

THE COMPUTER SYSTEMCENTER
3330 Piedmont Road
ATLANTA, GA 30305
(404)-231-1691

CHICAGO COMPUTER STORE
517 Talcott Rd
PARK RIDGE, IL 60068
(312)-823-2388

THE COMPUTER STORE
OF ANN ARBOR
310 East Washington Street
ANN ARBOR, MI 48104
(313)-995-7616

THE COMPUTER ROOM
3938 Beau D'Rue Drive
EAGAN, MN 55122
(612)-452-2567

GATEWAY ELECTRONICS, INC
8123-25 Page Blvd
ST. LOUIS, MO 63130
(314)-427-6116

ALTAIR COMPUTER CENTER
611 N. 27th St. Suite 9
LINCOLN, NB 68503
(402) 474-2800

COMPUTER SHACK
3120 San Mateo N E
ALBUQUERQUE, NM 87110
(505)-883-8282, 883-8283

THE COMPUTER STORE
269 Osborne Road
ALBANY, NY 12211
(518)-459-6140

THE COMPUTER STORE
OF NEW YORK
55 West 39th Street
NEW YORK, NY 10018
(212)-221-1404

COMPUTER STORES
OF CAROLINA, INC
1808 E. Independence Blvd
CHARLOTTE, N.C. 28205
(704)-334-0242

ALTAIR COMPUTER CENTER
5252 North Dixie Drive
DAYTON, OHIO 45414
(513)-433-8460

ALTAIR COMPUTER CENTER
110 The Annex
5345 East Forty First St
TULSA, OK 74135
(918)-664-4564

ALTAIR COMPUTER CENTER
8105 SW Nimbus Ave
BEAVERTON, OR 97005
(503)-644-2314

ALTAIR COMPUTER CENTER
5750 Bintliff Drive Suite 206
HOUSTON, TX 77036
(713)-780-8981

COMPUTERS-TO-GO
4503 West Broad St
RICHMOND, VA 23230
(804)-355-5773

MICROSYSTEMS (Washington, D.C.)
6605A Backlick Rd
SPRINGFIELD, VA 22150
(703)-569-1110

THE COMPUTER STORE
Suite 5
Municipal Parking Building
CHARLESTON, W. VA. 25301
(304)-345-1360

mits